








# **HY-IDE Software User's Manual**

## Table of Contents

<b>1. HY-IDE OVERVIEW .....</b>	<b>4</b>
1.1 INTRODUCTION .....	4
1.2 HY-IDE INSTALLATION AND SYSTEM REQUIREMENT .....	4
1.3 INSTALLATION AND UNINSTALLATION .....	4
1.3.1. Installation .....	4
1.3.2. Uninstallation .....	7
1.4 REGISTER .....	7
1.5 USE DEMO CODE TO GUIDE USER MANUAL .....	8
<b>2. HY-IDE INTERFACE DESCRIPTION .....</b>	<b>10</b>
2.1 HY-IDE EDIT INTERFACE .....	10
2.1.1 Edit Window .....	11
2.1.2 File Menu .....	11
2.1.3 Edit Menu .....	12
2.1.4 View Menu .....	12
2.1.5 Edit & Execution Menu .....	13
2.1.6 Options Menu .....	13
2.1.7 Window .....	16
<b>3. HY-IDE DEBUG INTERFACE.....</b>	<b>17</b>
3.1 FAST EXECUTION .....	18
3.2 RAM WINDOW .....	20
3.3 REGISTER WINDOW .....	23
3.4 WATCH WINDOW .....	25
3.5 STACK WINDOW .....	26
3.6 ADC WINDOW .....	27
3.7 OP WINDOW .....	32
3.8 COMPARATOR WINDOW .....	34
3.9 REGISTER & SRAM REVISE RECORD .....	36
3.10 HINT FUNCTION OF SOURCE CODE WINDOW .....	39
<b>4. PROGRAMMING WINDOW .....</b>	<b>41</b>
4.1 INTERFACE SETTING .....	41
4.2 OPERATION PROCEDURES .....	43
4.2.1 Open File and Assemble.....	44
4.2.2 Download Hex File to Programmer or IDE Flash Memory .....	47
4.3 PC ONLINE OTP PROGRAMMING.....	48
4.3.1 Blank Check  .....	49

4.3.2	Program 	49
4.3.3	Verify Program 	50
4.3.4	Read 	50
4.3.5	AUTO 	51
<b>4.4</b>	<b>OFFLINE PROGRAMMING</b>	<b>52</b>
4.4.1	Program Description	52
4.4.2	Program Times Restriction	53

**Attention :**

1. Information included in this manual concerning device applications and analogous content is furnished only for user convenience and HYCON does not accountable for the content usage and any incurred outcome. The content of specification may be replaced by updates; it is users' responsibility to assure that your application meets with your requirements and standards.
2. Should buyers applied HYCON products as key devices in life support or safety system is entirely at buyers' risk. Any use of HYCON products in such application is forbidden. HYCON reserves the right to change without further notice. For most up-to-date information, please visit our website:

<http://www.hycontek.com>

## 1. HY-IDE Overview

### 1.1 Introduction

To facilitate the process of product development, HY-IDE development environment is provided to deploy HYCON's full range of MCUs. Customers can implement the end-product's in-circuit emulation on this platform and program the code onto HY series' OTP products.

### 1.2 HY-IDE Installation and System Requirement

**The minimum requirement of system configuration to operate HY-IDE :**

#### 1. PC hardware request

PC compatible machine with PENTIUM® CPU  
128 MB Memory ( 256MB is recommended )  
10 GB Hard Disk Space

#### 2. OS

Windows 98SE  
Windows 2000  
Windows XP

#### 3. Applicable interface

USB Port

### 1.3 Installation and Uninstallation

#### 1.3.1. Installation

Note that some Windows operating system may request the HY-IDE to be installed in the Supervisor Privilege.

- Insert the HY-IDE CD into the CD ROM drive and find the file in the CD ROM or file to execute Setup.exe.
- Following the dialog step by step to continue setup procedures. As shown in
- Figure 1-1.
- First-time installation must initiate USB driver program, the setup procedures are as Figure 1-2 shown.

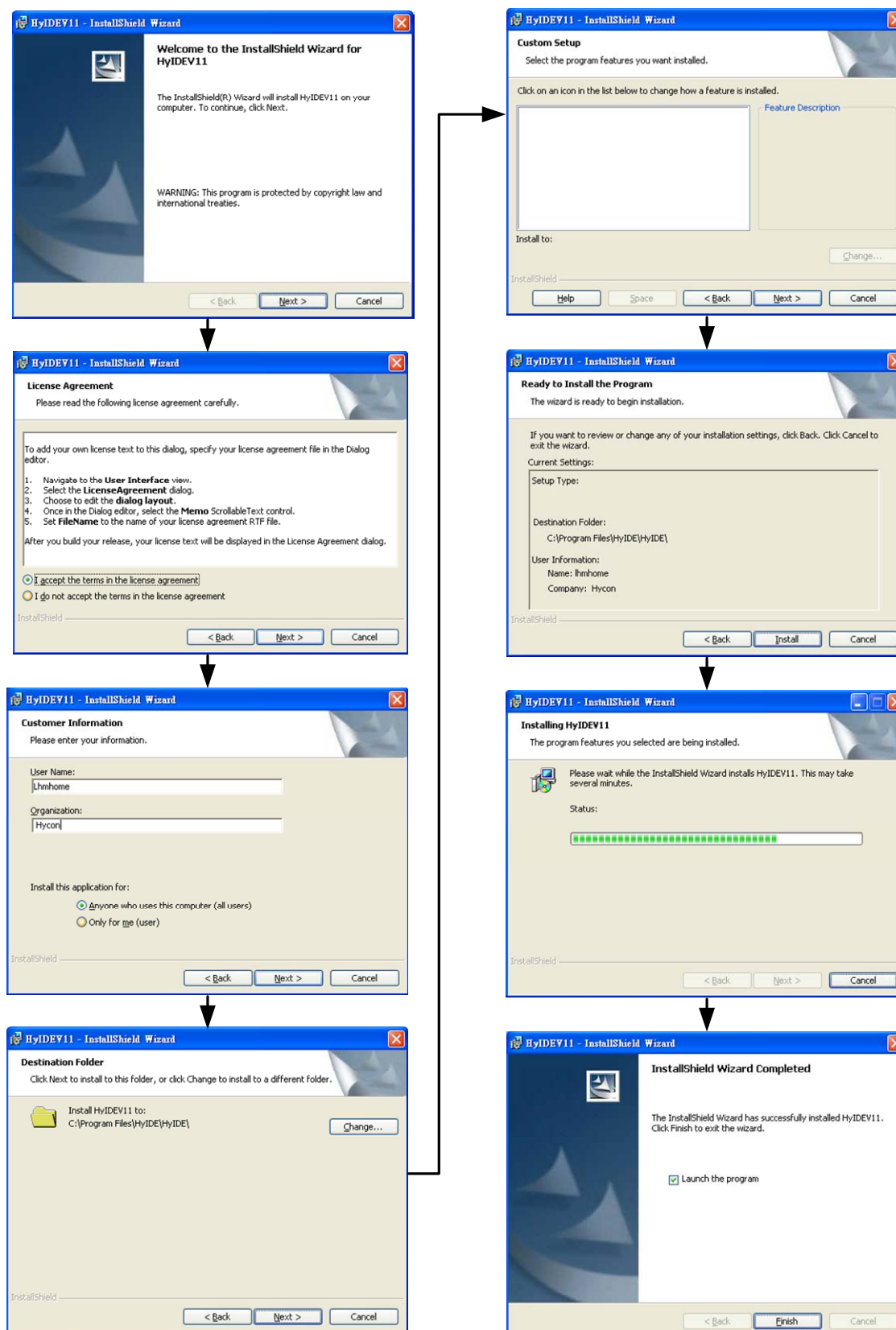


Figure 1-1



Figure 1-2

## 1.3.2. Uninstallation

Please remove the file "HyIDEV11" in "Add/Remove" under Control Panel.

## 1.4 Register

1. Customers use ICE hardware emulation or OTP programming for the first time.
2. If the dialog appeared or abnormal IDE crash occurred, customers must conduct re-register action.

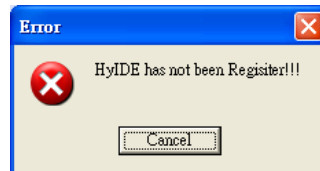


Figure 1-3

## Register Procedures

1. Please check the HyIDE Machine Number (HyIDE Code) on the parcel and send the number by e-mail or on-line register. HYCON will send back another customer register code to you.  
Connect the HyIDE Control Board to PC through USB interface.
2. Execute HyIDEV10 software (HyIDE.exe). Go to "Option" and press "Register".
3. Fill in the customer code in "Register Number" and click "Write" to start.

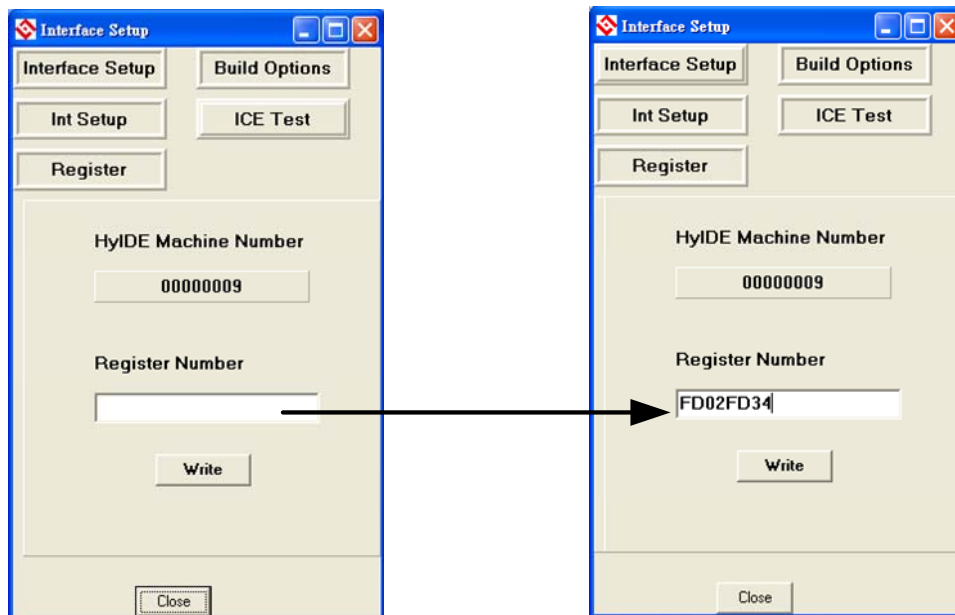


Figure 1-4

4. If the process is successful, a dialog will be shown.

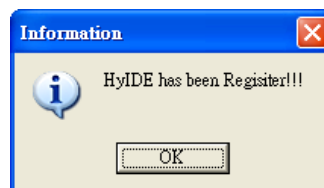


Figure 1-5

5. If the process is fail, a dialog will be shown as Figure 1-6.



Figure 1-6

6. Once the register is succeeded, Customers do not have to worry that other numbers may be written into the "Register Number".

## 1.5 Use Demo Code to Guide User Manual

- Start KitchFor11P13.asm
- Set the file as assembly main file
- Assembly start to progress program debug

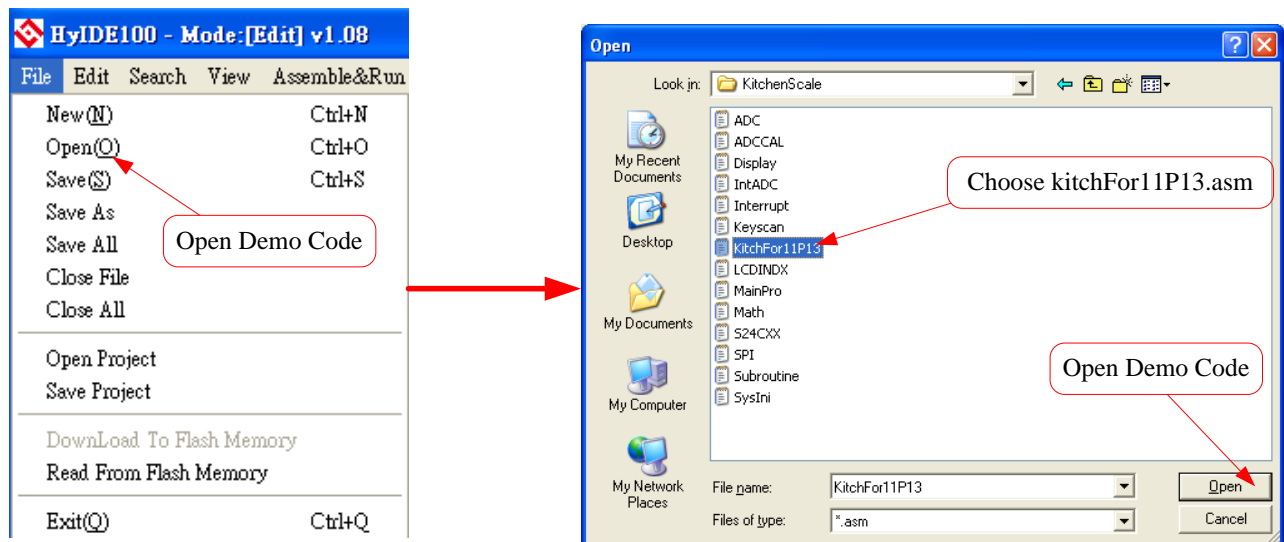


Figure 1-7



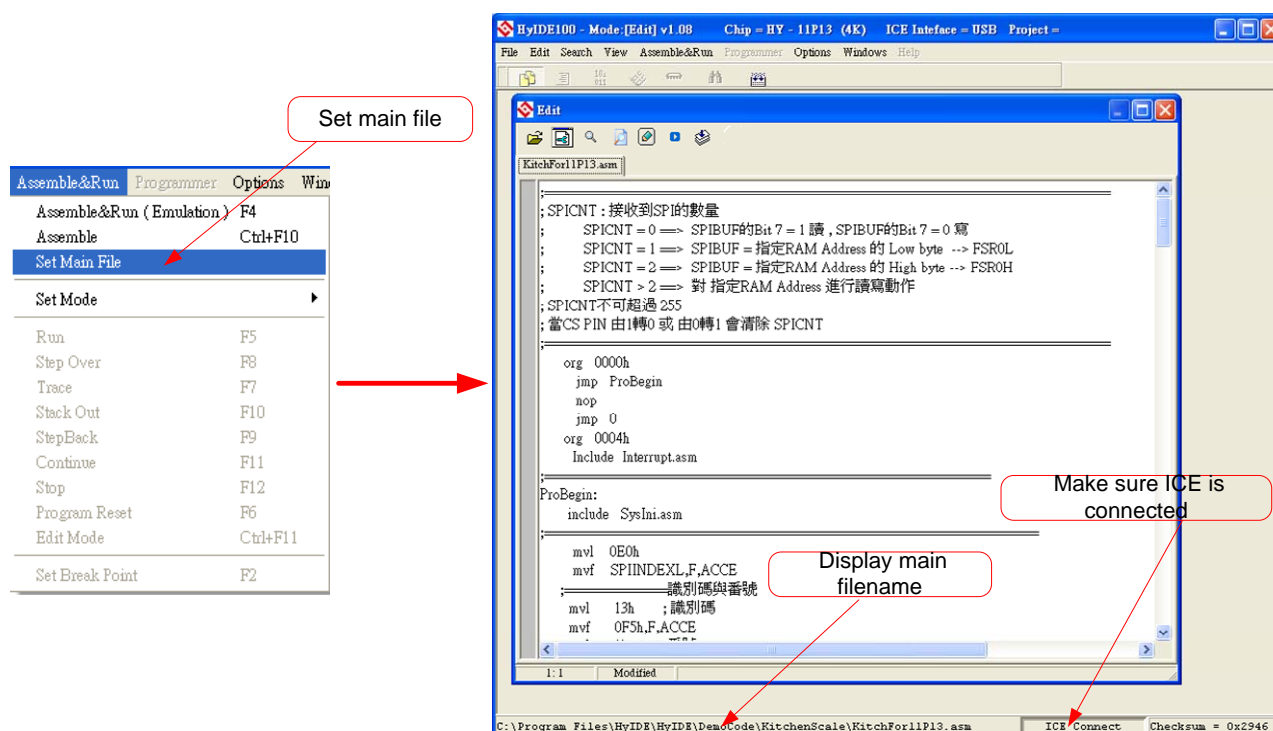


Figure 1-8

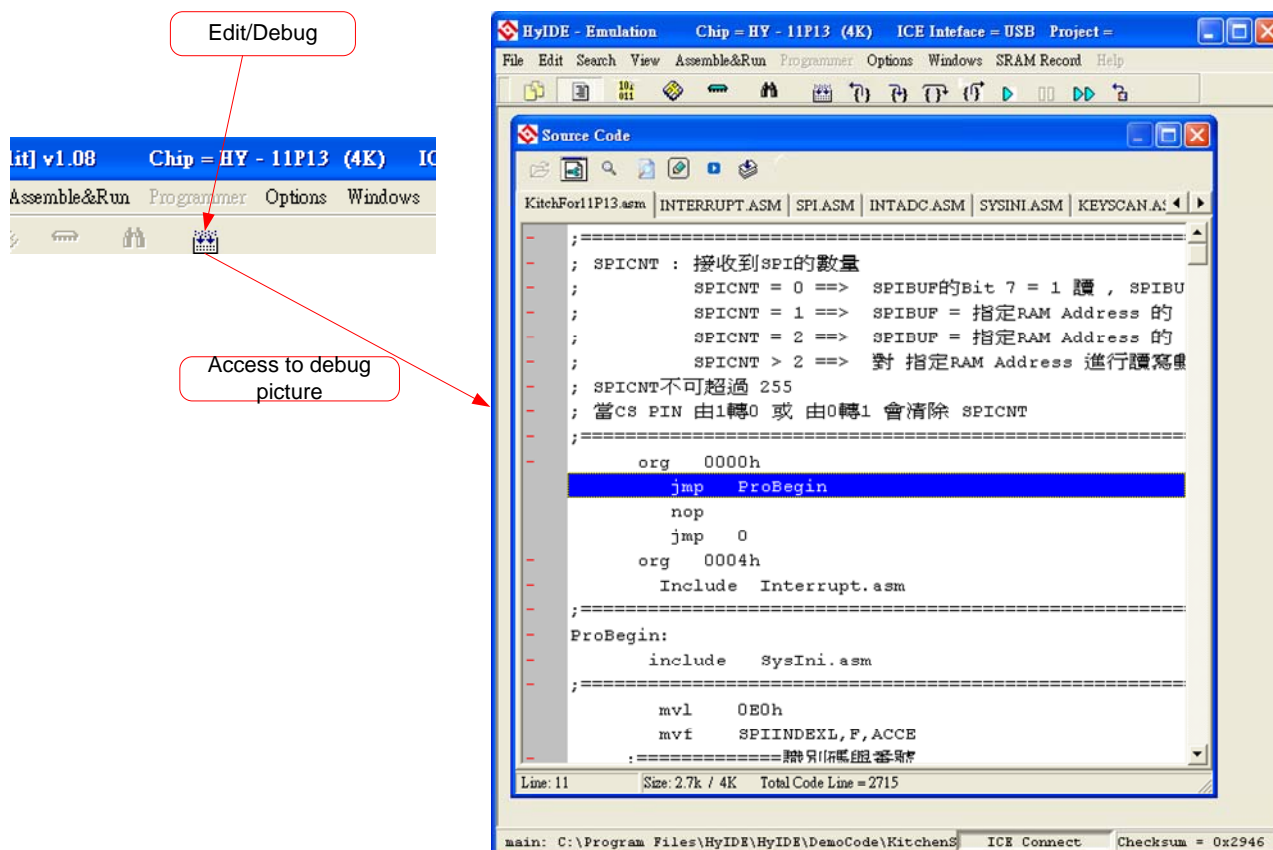


Figure 1-9

Any editor can be used to edit Source Code, as long as it can be stored as ASCII Code format.  
Debug and edit function will be elaborated in next Chapter.

## 2. HY-IDE Interface Description

### 2.1 HY-IDE Edit Interface

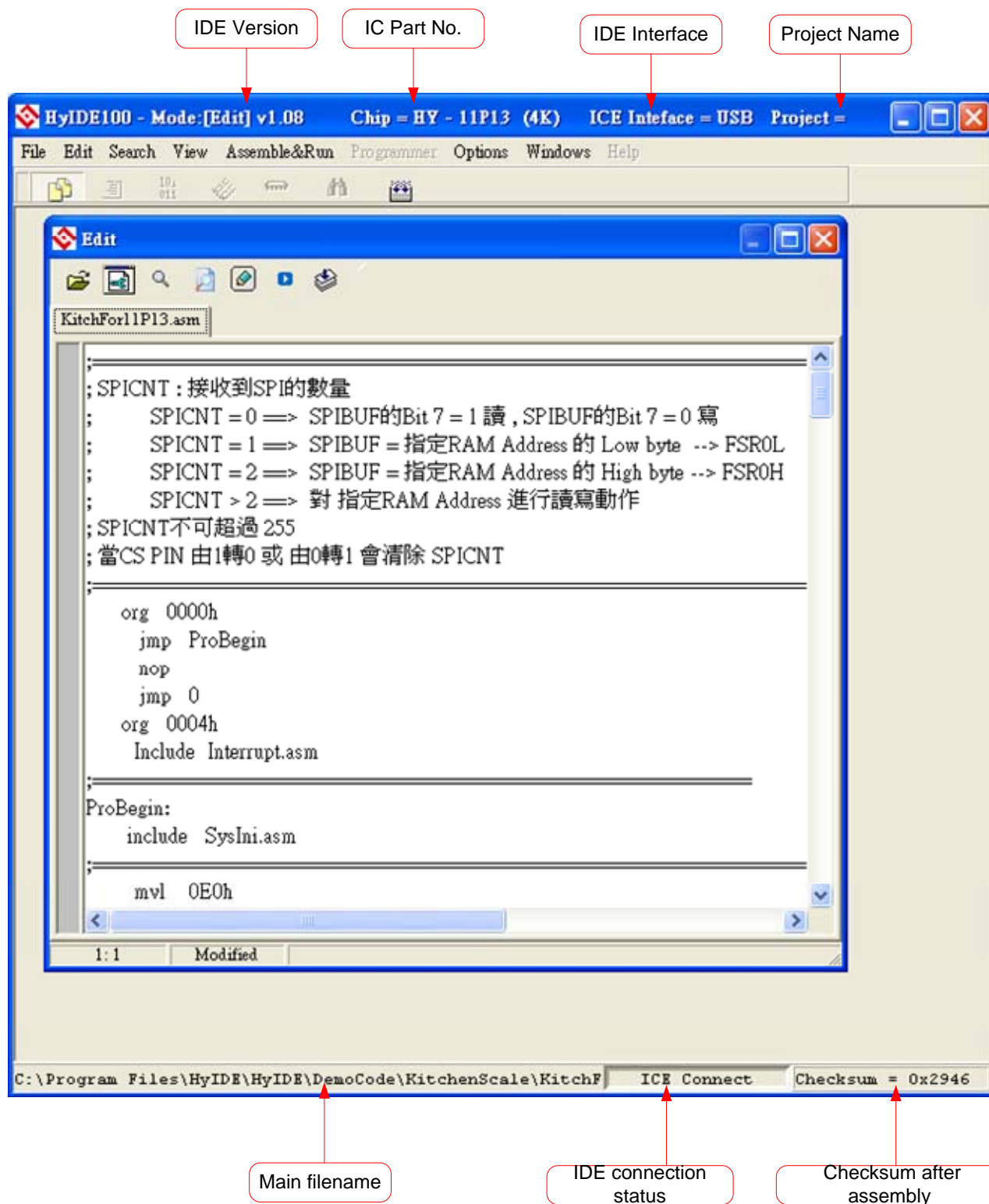









Figure 2-1

- New  
Create a new file
- Open  
Open an existing file
- Close  
Close the current active file

## 2.1.1 Edit Window

- Open   
Open an existing file.
- Label Setting   
Set label. When open too many files, this icon helps to return to the label setting place quickly.
- Jump to Label   
Jump to where to label has been placed.
- Find   
Search the entered word.
- Search the specified word. 
- Switch Display Window   
When too many files are opened, this icon helps to command file switching.
- Edit   
Only editor is executed. It will not get into program debug status.  
The information dialog may pop up after edit accomplished.



## 2.1.2 File Menu

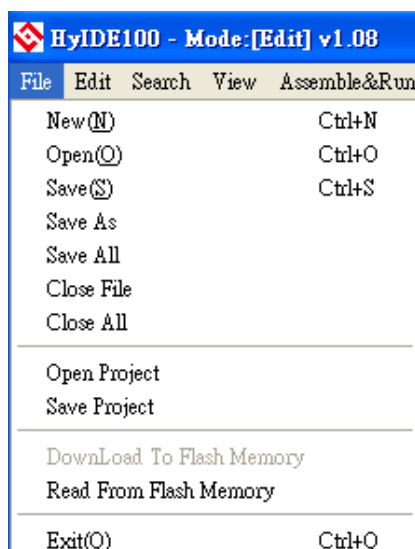


Figure 2-2

- New → Create a new file.
- Open → Open an existing file.
- Save → Write the active window data to the active file.
- Save As → Write the active window data to the specified file.
- Save All → Write all windows data to the corresponding opened files.
- Open Project → Project includes IC part no., IDE interface, Edit main filename, Active opened status, and Checksum. Project status will be loaded in after the project is opened.
- Save Project → Write the active project to the active project file.
- Close → Close the current active HyIDE program.

## 2.1.3 Edit Menu

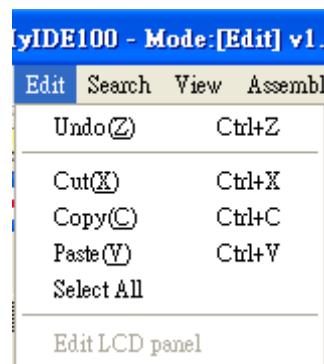


Figure 2-3

Under editing the file:

- Undo → Cancel the previous editing operation.
- Cut → Remove the selected lines from the file.
- Copy → Place a copy of the selected lines.
- Paste → Paste the copy lines to the present insertion point.
- All → Select all lines in the active file.

## 2.1.4 View Menu

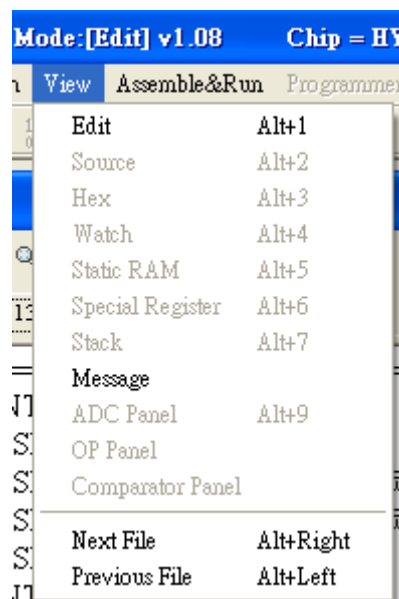


Figure 2-4

- Edit Window → Appoint the edit window as the present active window.
- Next file → Appoint the next file as the present active window.
- Previous file → Appoint the previous file as the present active window.

## 2.1.5 Edit & Execution Menu

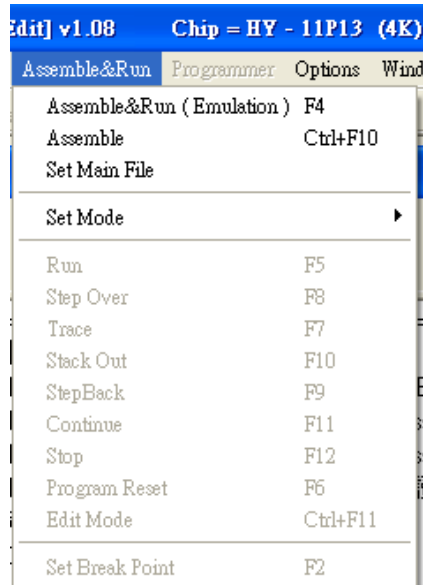


Figure 2-5

- Edit & Execution → Edit Source Code and execute program debug mode.
- Edit → Only program is edited, program debug is not executed. This assembler will not generate error message according to IC part no. Error message will show up when the lines is error. It is usually used in generating OBJ Code (Object).
- Set Edit Main File → Set the file as edit main file. Files will be named after compiler generated file name, such as Hex, MAP, ASC...etc.
- Debug → Debug through software or hardware is selective.

## 2.1.6 Options Menu

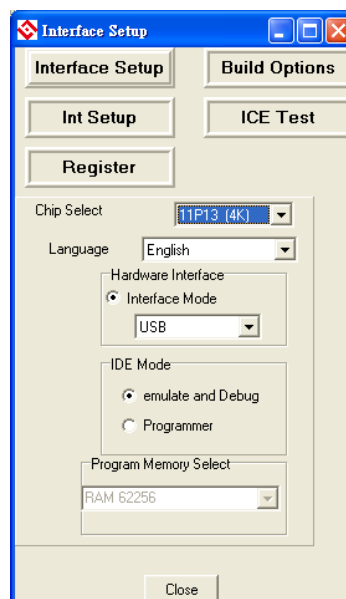


Figure 2-6

There are five options:

(1) Interface Setting

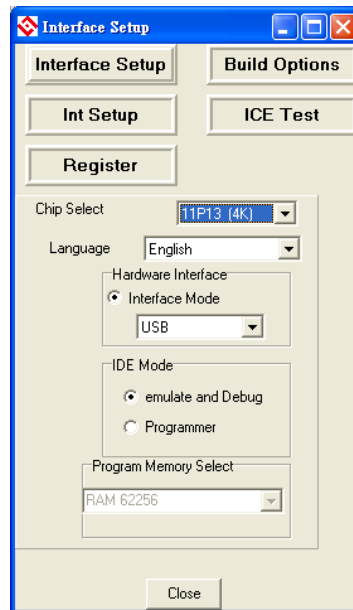


Figure 2-7

- IC option: Select IC part no. Compiler will assemble the selected part no.'s program file. It will determine whether there is any misuse or non-existing Register or SRAM, or has the program exceeded the ROM Size.
- Language option: English and Chinese interface are selectable.
- Communication interface option: Select IDE communication interface.
- Mode option: Two choices, Emulate and debug and program.

(2) Edit Item

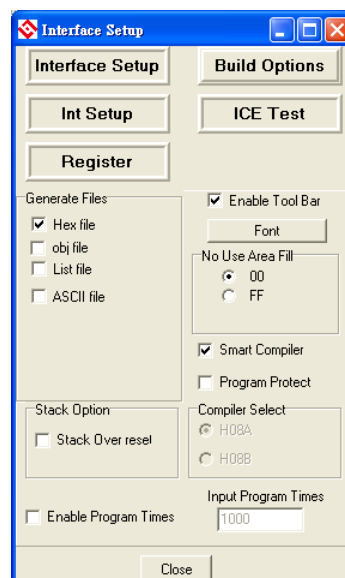


Figure 2-8

- Assembler generated extension: it is selectable to produce below file format.
  1. binary file : Hex
  2. obj file : obj
  3. List file : lst
  4. ADCII file : asc
- Stack operation: Choose to replace the program after stack overflow. When this option is chosen, Compiler will add to Hex, it will be programmed in to OTP.
- Program number of times limit: Please refer to the program Chapter.
- Font option: Choose editor's fonts.
- Fill unused zone: Fill the unused zone with 0x0000 or 0xFFFF in the program.
- Simplified assemble: Simplified assemble function is selectable. When JMP or CALL is smaller than 2K, it will automatically transform to RJ or RCALL. If the arguments of CALL are set, it will not transform to RCALL.
- Program protection: Please refer to the program Chapter.

### (3) Interrupt Setting

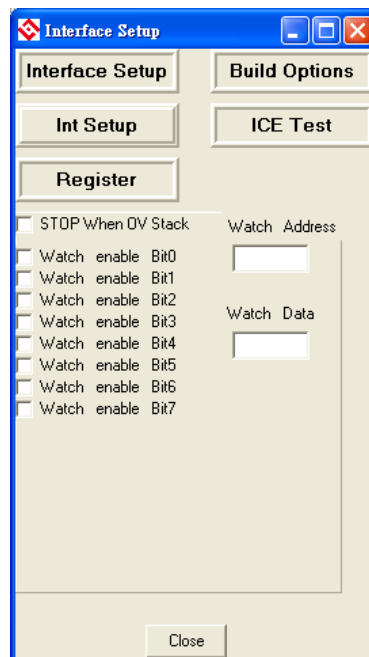


Figure 2-9

- Stop operation when Stack overflow: IDE will stop when Stack overflow.
- Monitor address: Select the monitored Register or RAM. The program will stop when the program executed RAM or Register value equals to the monitored Data.
- Monitor Data: Monitor value is set when the monitor Data is filled up.
- Monitor RAM bit: Monitor function will be activated if the monitor bit is marked on. The program will stop when the bit of Data value equals to the marked on bit.

## (4) ICE Test

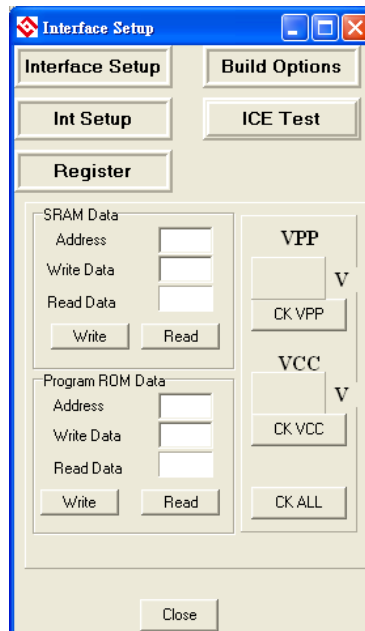


Figure 2-10

## 2.1.7 Window

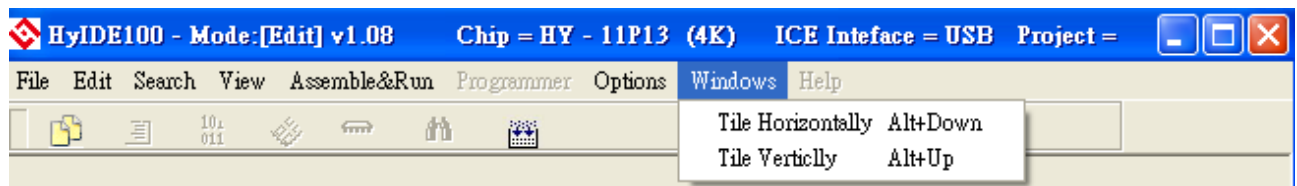


Figure 2-11

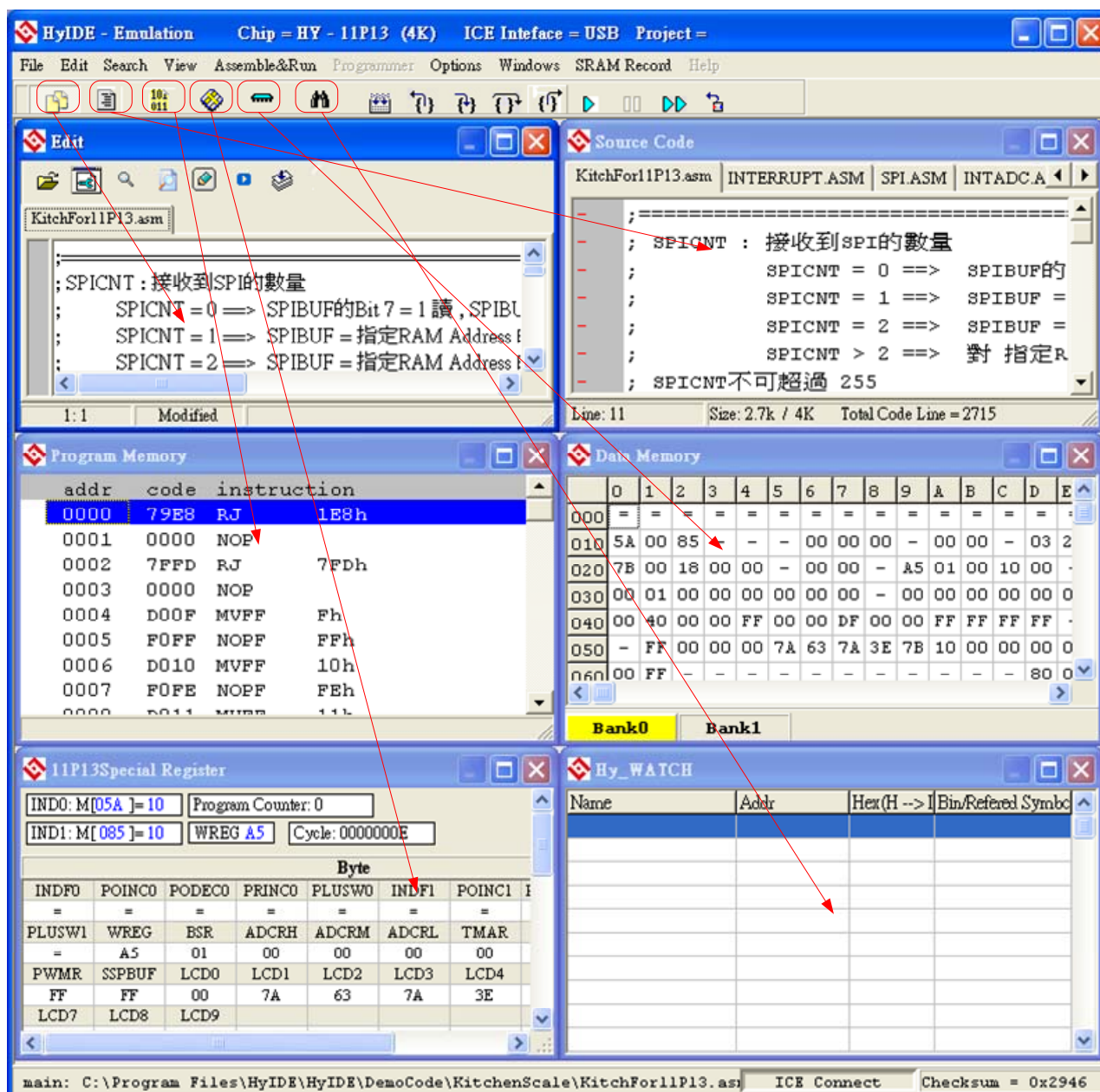
The window can be displayed horizontally or vertically.



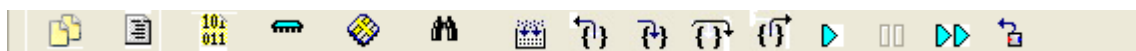
## 3. HY-IDE Debug Interface

It can be classified into hardware debug and software debug.

- Hardware debug  
The indication column is blue
- Software debug  
The indication column is green



## 3.1 Fast Execution



### Fast Window Switch

(2) Switch to Edit window

(3) Switch to Source window

(4) Switch to Hex window

addr	code	instruction
0000	79E8	RJ 1E8h
0001	0000	NOP
0002	7FFD	RJ 7FDh
0003	0000	NOP
0004	D00F	MVFF Fh
0005	F0FF	NOFF FFh
0006	D010	MVFF 10h
0007	F0FE	NOFF FEh
0008	D011	MVFF 11h

(5) Switch to Ram window

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E
000	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=
010	5A	00	85	-	-	-	00	00	00	-	00	00	-	03	2
020	7B	00	18	00	00	-	00	00	-	A5	01	00	10	00	-
030	00	01	00	00	00	00	00	00	-	00	00	00	00	00	0
040	00	40	00	00	FF	00	00	DF	00	00	FF	FF	FF	FF	-
050	-	FF	00	00	00	7A	63	7A	3E	7B	10	00	00	00	0
060	00	FF	-	-	-	-	-	-	-	-	-	-	-	80	0

(6) Switch to Reg window







Byte							
INDF0	POINCO	PODECO	PRINCO	PLUSW0	INDF1	POINC1	I
=	=	=	=	=	=	=	=
PLUSW1	WREG	BSR	ADCRH	ADCRM	ADCRL	TMAR	
=	A5	01	00	00	00	00	
PWMR	SSPBUF	LCD0	LCD1	LCD2	LCD3	LCD4	
FF	FF	00	7A	63	7A	3E	
LCD7	LCD8	LCD9					

(7) Switch to Watch window

Name	Addr	Hex(H --> I)	Bin/Refered Symbol

### Fast Debug

- (1) Step back
- (2) Trace (Enter into Macro/vice program)
- (3) Step over (Not enter into Macro/vice program)

- (4) Skip Call 
- (5) Execute (Free RUN) 
- (6) Pause 
- (7) Continue 
- (8) Program replace 
- (9) Back to edit mode 

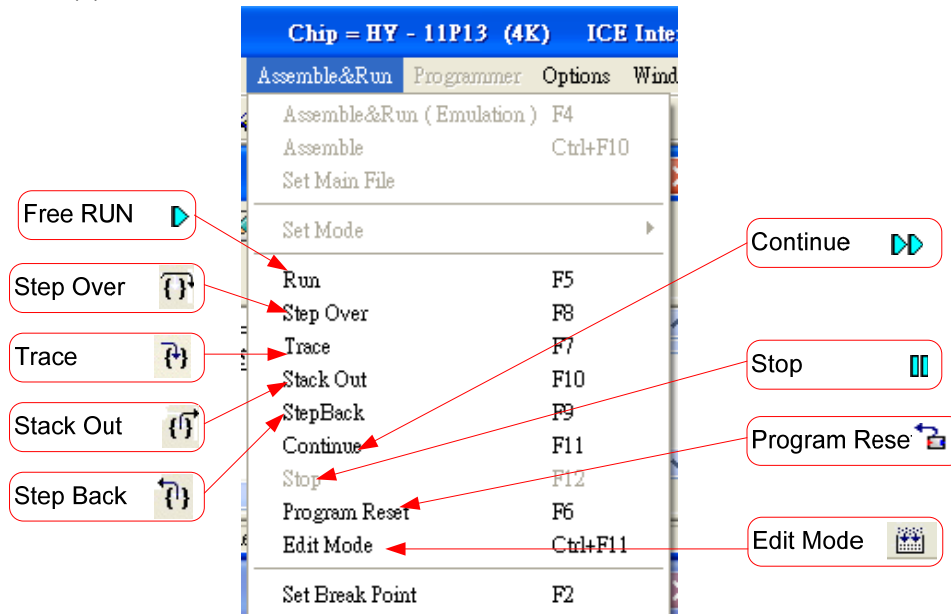


Figure 3-1

Two methods to set or remove interrupt:

1. Use mouse to select interrupt place in program code window or machine code window, press "F2" button to set to remove interrupt.
2. Use mouse to select interrupt place in program code window or machine code window, double click the left key to set or remove interrupt

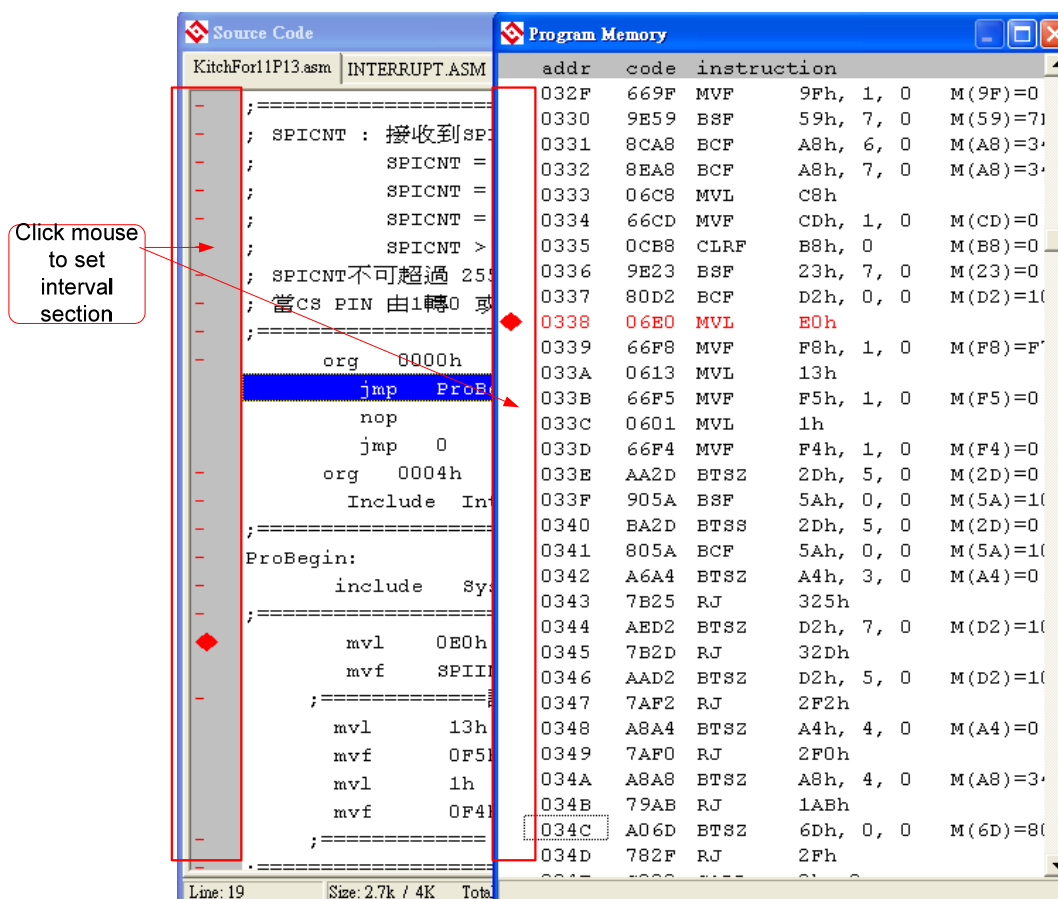


Figure 3-2

### 3.2 RAM Window

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	00
010	5A	00	85	-	-	-	00	00	00	-	00	00	-	03	2E	7F
020	7B	00	18	00	00	-	00	00	-	A5	01	00	10	00	-	-
030	00	01	00	00	00	00	00	00	-	00	00	00	00	00	00	00
040	00	40	00	00	FF	00	00	DF	00	00	FF	FF	FF	FF	-	30
050	-	FF	00	00	00	7A	63	7A	3E	7B	10	00	00	00	03	-
060	00	FF	-	-	-	-	-	-	-	-	-	-	-	80	00	00
070	00	00	00	-	1C	00	-	00	00	00	-	-	-	-	-	-
080	A5	A5	A5	A5	00	10	09	02	03	04	03	00	7B	00	00	00
090	00	00	00	00	00	29	43	03	00	00	00	00	00	00	00	00
0A0	00	00	00	00	00	00	01	00	34	00	00	00	00	07	EF	DE
0B0	79	DE	00	00	00	00	A0	FB	00	00	00	00	00	00	00	00
0C0	00	00	00	00	01	00	00	00	00	00	00	00	00	5E	00	00
0D0	00	00	10	FD	D8	79	DE	D8	79	DE	BB	79	DE	0A	7A	DE
0E0	EF	79	DE	A8	79	DE	0C	7A	DE	D4	79	DE	00	04	00	00
0F0	08	00	E7	79	00	00	FF	00	F7	00	00	00	85	00	5A	00

Bank0 Bank1

Figure 3-3

After opening RAM window, Bank will show the volume of the selected IC. Every Bank has 256 byte.

Bank0 starts from 0x00 to 0xFF. Bank1 starts from 0x100 to 0x1FF...etc.

If the address does not exist, it will display "-".

If users intend to switch Bank display, use cursor to point to the desired Bank zone, and then click the

left key of the mouse to confirm.

If Hint is set, the address will display numbers and will be underlined.

**Notice: The Address 0x00 ~ 0x0Eof Bank0 is indirect addressing register, it cannot be revised directly, the displayed value is not referable. If revise is required, please refer to Chapter 3.3: Revise indirect addressing Data or Address.**

## Function Display

Click the mouse selection key (right key)

<u>S</u> et Mark	
S <u>e</u> t Mark(new color)	
<u>R</u> eset Mark	
R <u>e</u> set All Mark	
<hr/>	
S <u>e</u> t <u>H</u> int	
R <u>e</u> set <u>H</u> int	
R <u>e</u> set <u>A</u> ll Hint	
<hr/>	
<u>L</u> oad RAM Data	
S <u>a</u> ve RAM Data	
S <u>a</u> ve <u>T</u> o excel	
R <u>A</u> M <u>B</u> A <u>N</u> K0	
R <u>A</u> M <u>B</u> A <u>N</u> K1	

Figure 3-4

- (1) Set Mark
- (2) Set Mark (new color)
- (3) Reset Mark
- (4) Reset All Mark
- (5) Set Hint
- (6) Reset Hint
- (7) Reset All Hint
- (8) Load RAM Data
- (9) Save RAM Data
- (10) Save To excel
- (11) RAMBANK0
- (12) ...

## Hint

Use DS defined SRAM; Hint will be automatically generated in corresponding window address. When cursor point to the address, it will show the defined string.

Ex: Program definition SRAM

MEMAR	080h	
MD1	DS	1
MD2	DS	1
MD3	DS	1
MDL1	DS	1
MDL2	DS	1
MDL3	DS	1

MD4	DS	5
S_REG	DS	1
r_Len	DS	1
SQRTmp	DS	4
Temp	DS	16

After assembling, it will enter into debug status, displaying memory window.

When cursor points to 80h address, <80>:MD1 will be shown.

When cursor points to 86h address, <86>:MD4[0] will show up.

When cursor points to 87h address, <87>:MD4[1] will show up.

-	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	00
010	5A	00	85	-	-	-	00	00	00	-	00	00	-	03	2E	7F
020	7B	00	18	00	00	-	00	00	-	A5	01	00	10	00	-	-
030	00	01	00	00	00	00	00	00	-	00	00	00	00	00	00	00
040	00	40	00	00	FF	00	00	DF	00	00	FF	FF	FF	FF	-	30
050	-	FF	00	00	00	7A	63	7A	3E	7B	10	00	00	00	03	-
060	00	FF	-	-	-	-	-	-	-	-	-	-	-	00	00	00
070	00	00	00	-	1C	00	-	00	00	00	-	-	-	-	-	-
080	A5	A5	A5	A5	00	10	09	02	03	04	03	00	7B	00	00	00
090	00	00	00	00	00	29	43	03	00	00	00	00	00	00	00	00
0A0	00	00	00	00	00	00	01	00	34	00	00	00	00	07	EF	DE
0B0	79	DE	00	00	00	00	A0	FB	00	00	00	00	00	00	00	00
0C0	00	00	00	00	01	00	00	00	00	00	00	00	5E	00	00	00
0D0	00	00	10	FD	D8	79	DE	D8	79	DE	BB	79	DE	0A	7A	DE
0E0	EF	79	DE	A8	79	DE	0C	7A	DE	D4	79	DE	00	04	00	00
0F0	08	00	E7	79	00	00	FF	00	F7	00	00	00	85	00	5A	00

Figure 3-5

There are two ways to revise SRAM value:

1. Point the cursor to the selected revised lines, click mouse's left key and Key IN directly.
2. Point the cursor to the selected revised lines, double click the mouse's left key, a window will pop up as Figure 3-6 shown. Users can key in on keyboard or press the button by mouse.

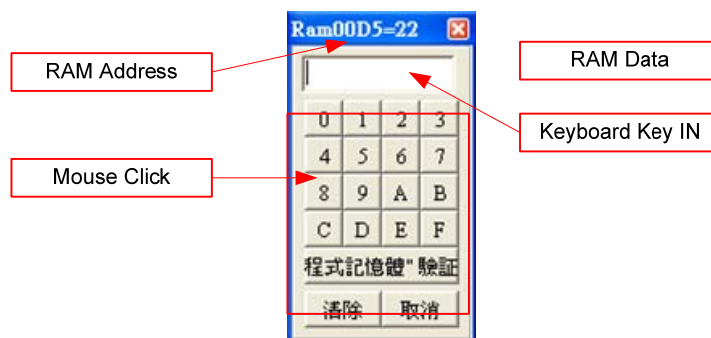


Figure 3-6

## 3.3 Register Window

The screenshot shows the '11P13Special Register' window. Annotations include:

- WREG**: Points to the WREG register in the Byte table.
- Program Counter**: Points to the 'Program Counter: 0' field.
- Cycle Times**: Points to the 'Cycle: 0000000E' field.
- Indirect addressing 0 Data**: Points to the 'IND0: M[05A]=10' field.
- Indirect addressing 0 Address**: Points to the 'IND0: M[05A]=10' field.
- Indirect addressing 1 Address**: Points to the 'IND1: M[085]=10' field.
- Indirect addressing 1 Data**: Points to the 'IND1: M[085]=10' field.
- Single Byte Register**: Points to the 'WREG' register in the Byte table.
- one Word composed Register**: Points to the 'FSR0' register in the Word table.
- Display PAGE 1 Register**: Points to the 'PAGE1' tab.
- Display PAGE 2 Register**: Points to the 'PAGE2' tab.
- Display PAGE 3 Register**: Points to the 'PAGE3' tab.
- Register byte**: Points to the 'BOR' register in the Page 1 table.
- Register bit**: Points to the 'ENRC2M' register in the Page 3 table.

INDF0	POINCO	PODEC0	PRINCO	PLUSW0	INDF1	POINC1	PODEC1	PRINC1
=	=	=	=	=	=	=	=	=
PLUSW1	WREG	BSR	ADCRH	ADCRM	ADCRL	TMAR	PRC	TMCR
=	A5	01	00	00	00	00	DF	00
PWMR	SSPBUF	LCD0	LCD1	LCD2	LCD3	LCD4	LCD5	LCD6
FF	FF	00	7A	63	7A	3E	7B	10
LCD7	LCD8	LCD9						
00	00	00						

FSR0	FSR1	TOS	PCLAT	TBLPTR	TBLD	PROD	TMBR	CCP0R
005A	0085	0000	0000	032E	7F7B	0018	FF00	FFFF
CCP1R								
FFFF								

STKPTR	STKFL	STKUN	STKOV	-	STKPRT3	STKPRT2	STKPRT1	STKPRT0
INTE1	GIE	ADCIE	TMCIE	TMBIE	TMAIE	WDTIE	ELIE	EOIE
INTE2	-	-	-	-	-	SSPIE	CCP1IE	CCP0IE
INTF1	-	ADCIF	TMCIF	TMBIF	TMAIF	WDTIF	ELIF	EOIF
INTF2	-	-	-	-	-	SSPIF	CCP1IF	CCP0IF
STATUS	-	-	-	C	DC	N	OV	Z
PTAUS	PD	TO	IDL	BOR	-	SKERR	-	-
LVDON	-	LVDIF	LVDOP	LVDON	VLDX3	VLDX2	VLDX1	VLDX0
PWRCN	ENVDDA	VDDAX1	VDDAX0	ENACM	-	-	-	-
MCKCN1	ADCS2	ADCS1	ADCS0	ADCCK	XTHSP	XTSP	ENXT	ENRC2M
MCKCN2	-	-	LSCK	HSCK	HSS1	HSS0	CPUCK1	CPUCK0
MCKCN3	LCD52	LCD51	LCD50	-	PERCK	BZS1	BZS1	BZS0

Figure 3-7

### ➤ Revise Indirect Addressing Data or Address

After setup as Figure 3-8 illustrated, Address can be revised through typing on the keyboard or by pressing the value by mouse.

The diagram illustrates the process to revise indirect addressing data or address. It shows the '11P13Special Register' window with the 'IND0: M[05A]=10' field. A red arrow points to the 'Modify Value' button, and another red arrow points to the 'Modify FSR0' button. A red arrow points to the 'click the left key of the mouse' annotation. A red arrow points to the 'FSR0=05A' window, which is a numeric keypad with buttons for digits 0-9, letters A-F, and a 'Memory checked OK' button. The 'Clear' and 'Cancel' buttons are also visible.

Figure 3-8

After setup as Figure 3-9 illustrated, Data can be revised through typing on the keyboard or by pressing the



value by mouse.

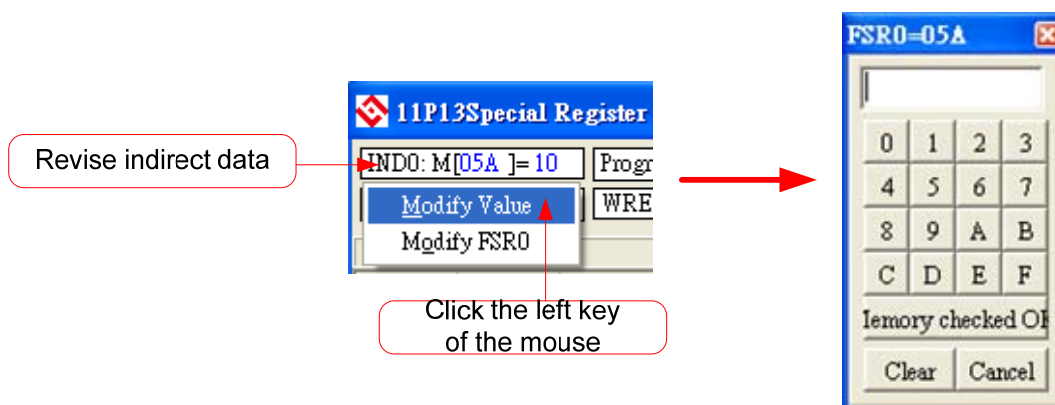


Figure 3-9

➤ Revise WREG Data

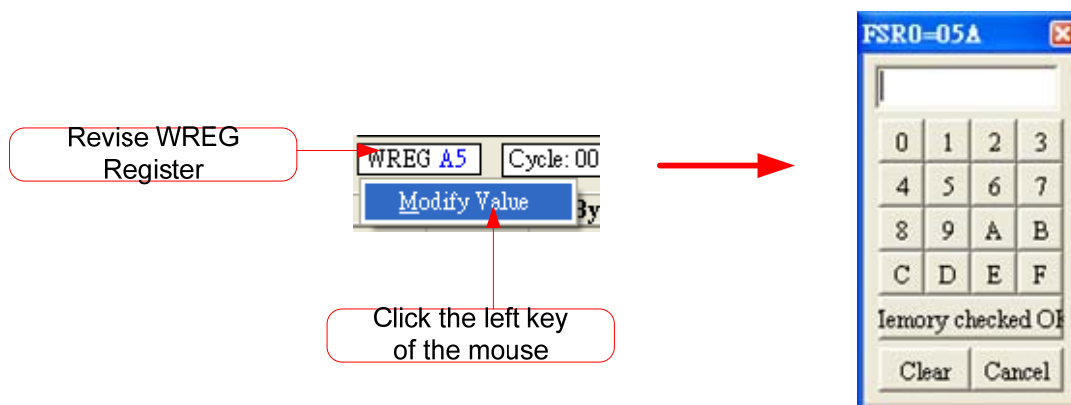


Figure 3-10

➤ Revise single 1 byte or Word Register Data

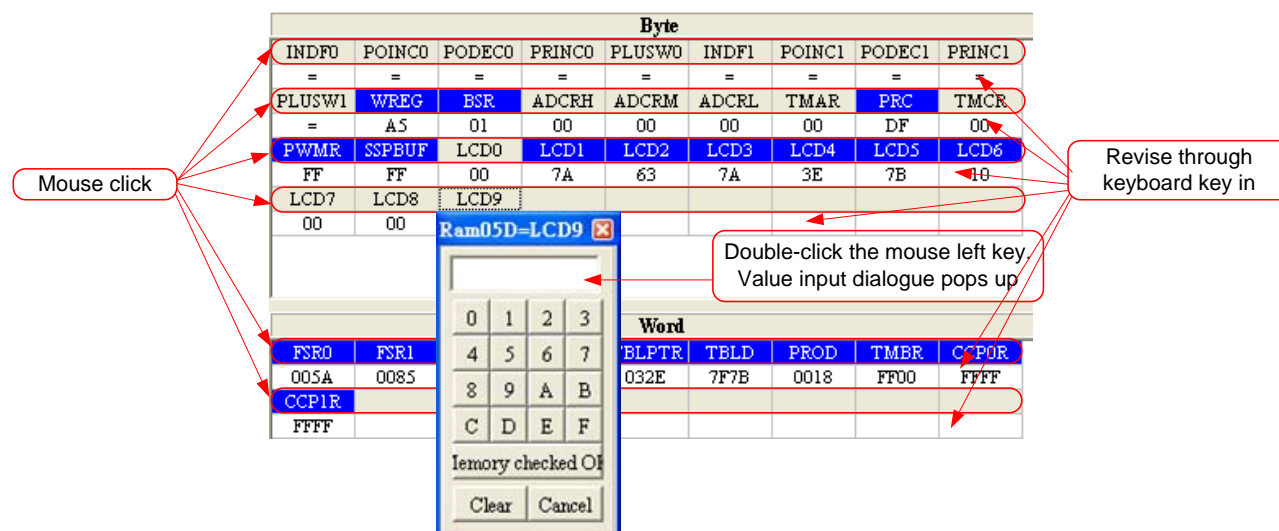


Figure 3-11

➤ Revise Register single 1 byte or single 1 bit

After Bit is configured as 1, its value will be highlighted in blue font.

After Bit is configured as 0, its value will be shown in black font.



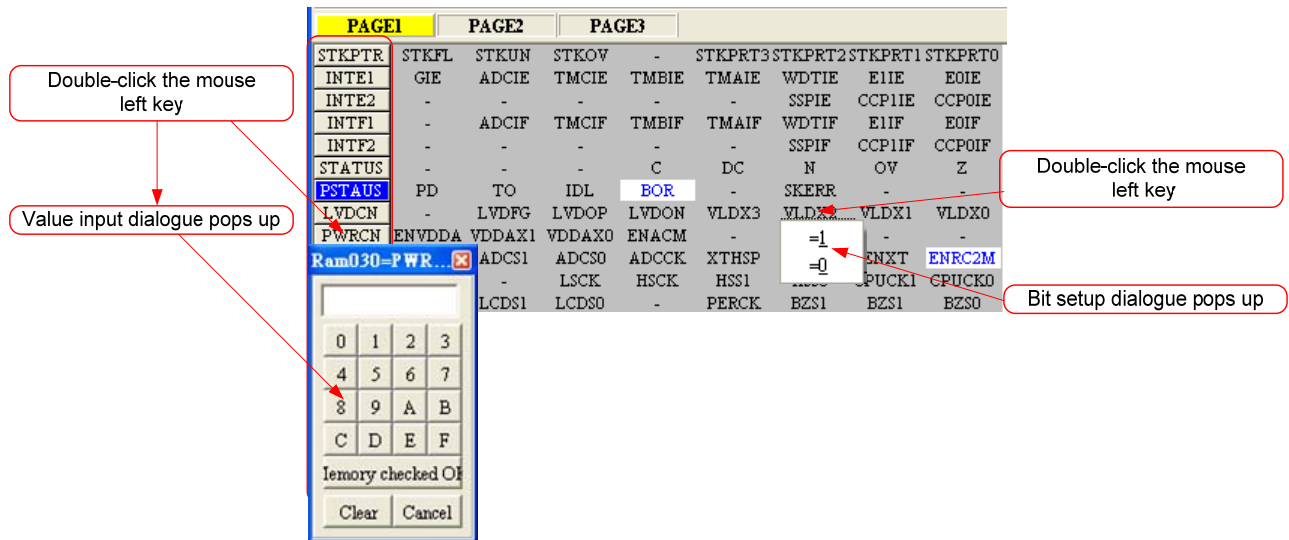


Figure 3-12

## 3.4 Watch Window

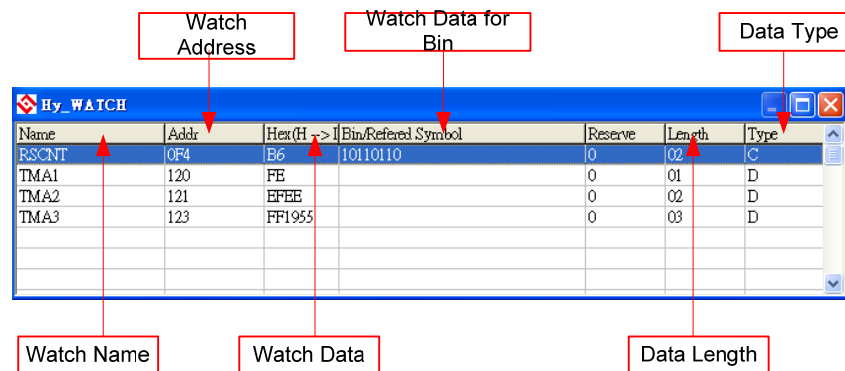


Figure 3-13

- Watch Name → Monitored Data name, program uses EQU or DS defined name.
- Watch Address → Monitored Data Address
- Watch Data → Reveal data. It is selectable to be arranged from right to left or from left to right. It can also display decimal or hexadecimal system.

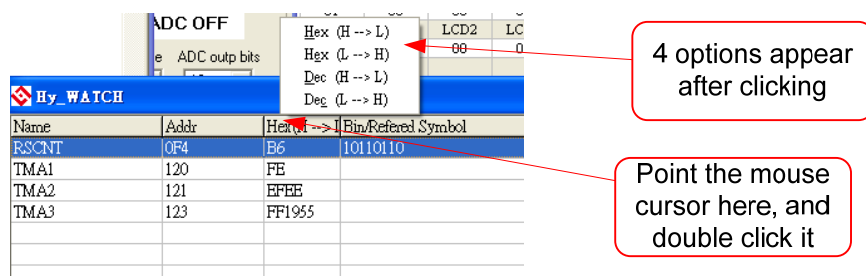


Figure 3-14

- Hex (H → L): Hexadecimal display, address H/L shows from low to high
- Hex (L → H): Hexadecimal display, address L/H shows from high to low
- Dec (H → L): Decimal display, address H/L shows from low to high
- Dec (L → H): Decimal display, address L/H shows from high to low

- Watch Data for Bin → Data display in binary system, only for those EQU defined Address.
- Data Length → Data length, showing DS definition length; if EQU definition is applied, this value will show "2".
- Data Type → Data type; D = DS definition; C = EQU definition.



Monitor EQU defined Register or RAM, click the right key of mouse to select add-in monitored Register or RAM as Figure 3-15 described.

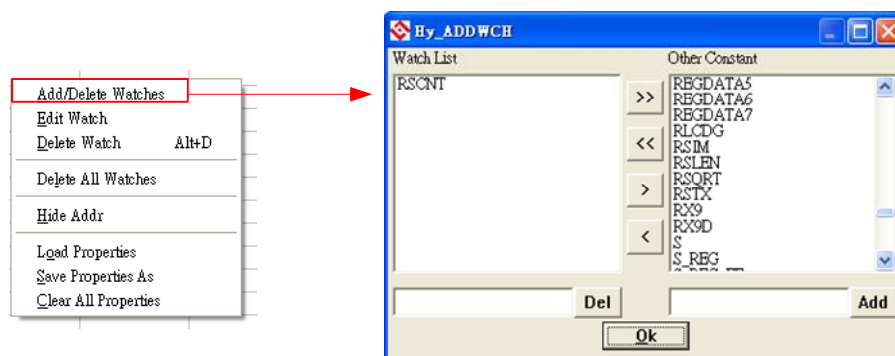


Figure 3-15

## 3.5 Stack Window

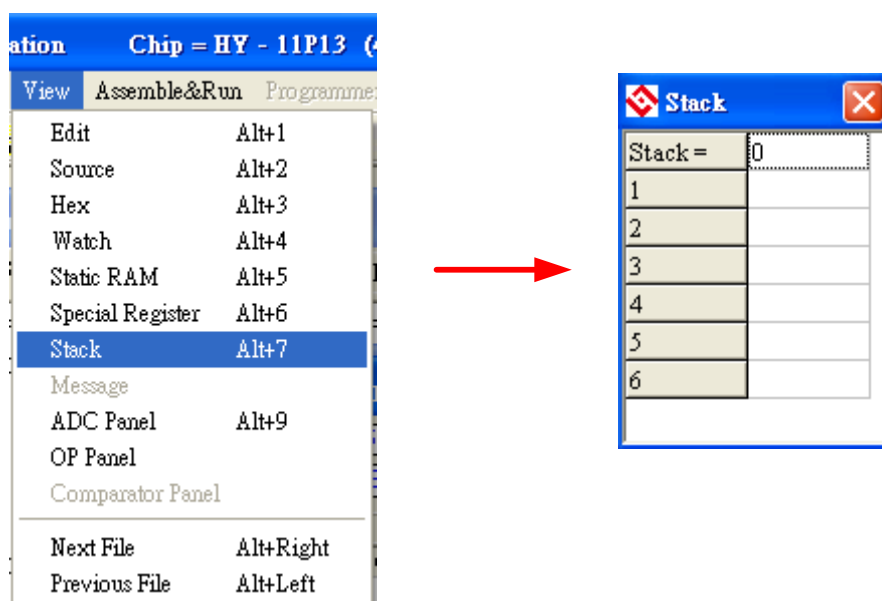


Figure 3-16

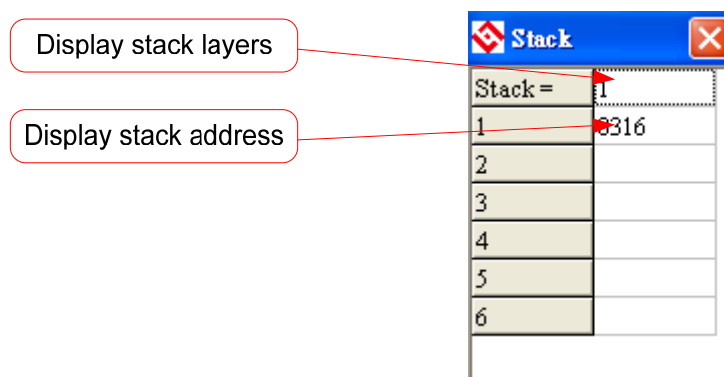


Figure 3-17

## 3.6 ADC Window

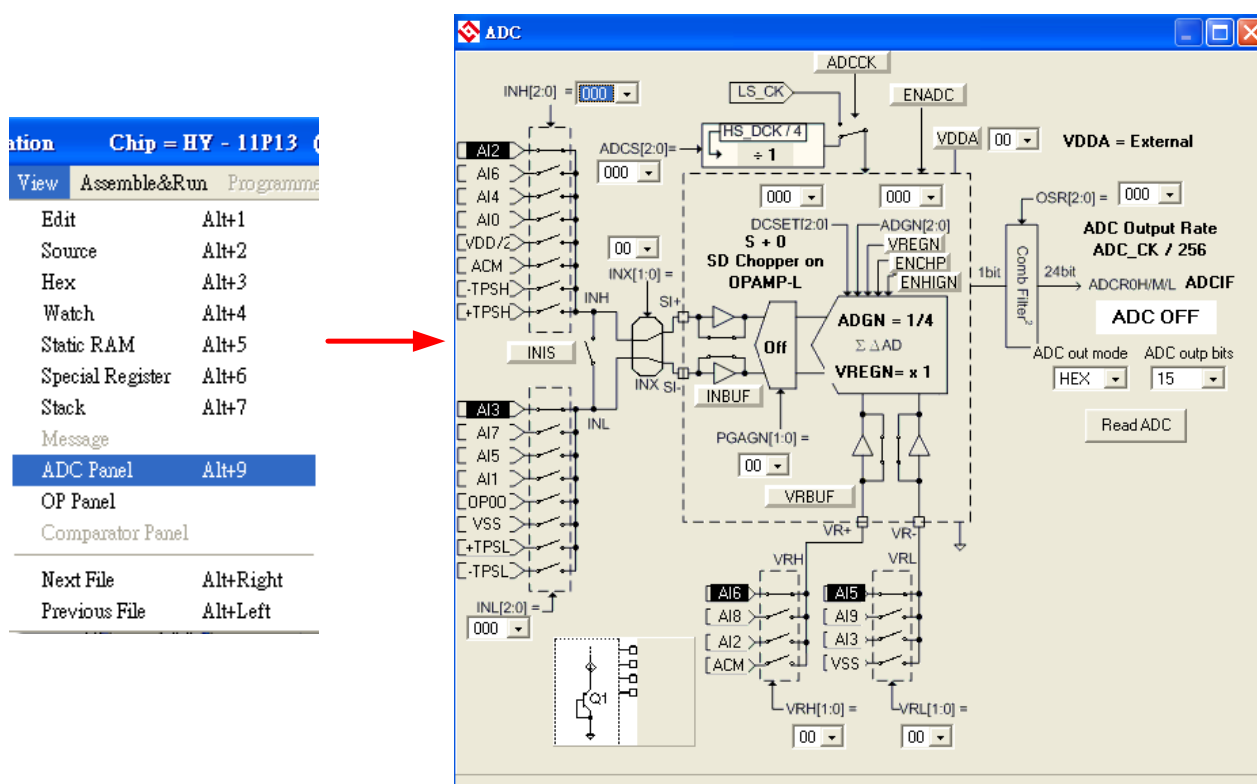


Figure 3-18

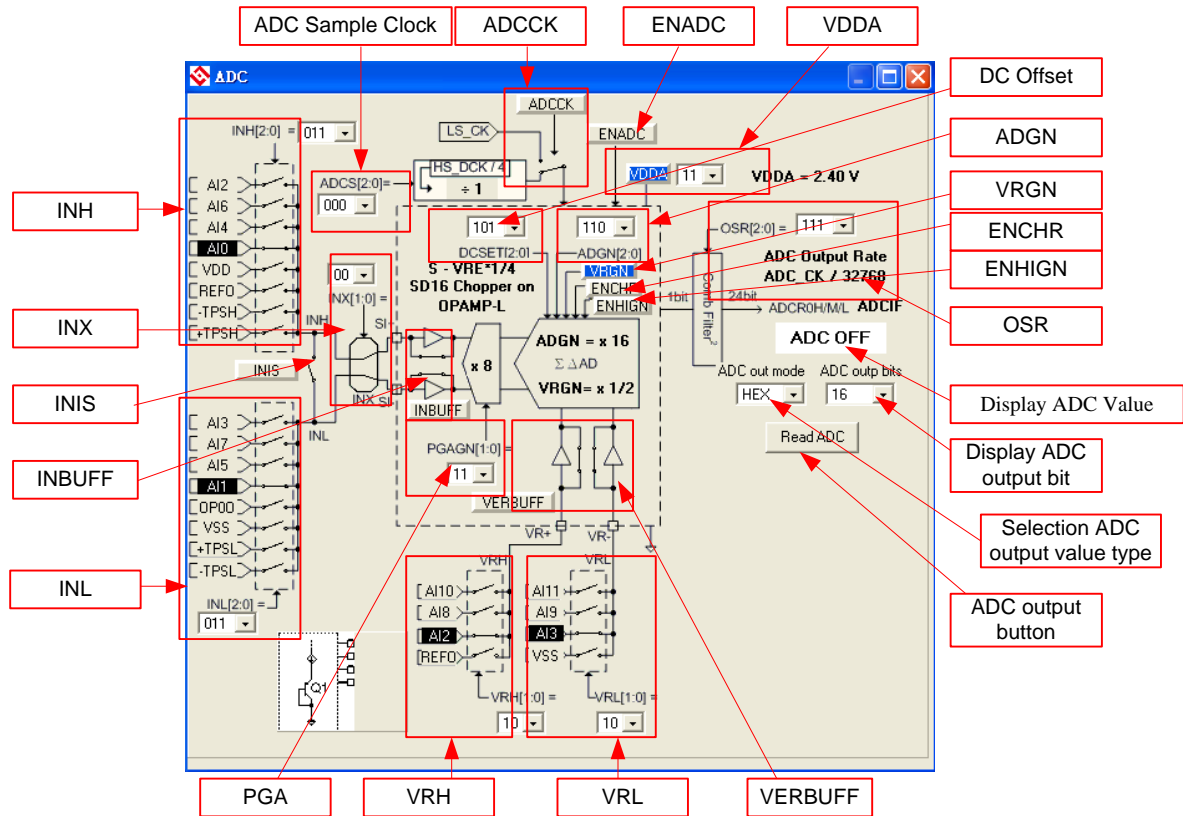


Figure 3-19



## INH Network

- (1) Click the network by mouse, INH can select the specified network.
- (2) Click the network switch by mouse, INH can select the specified network.
- (3) Click the mouse, a menu as Figure 3-20 will appear and users can select the switch network.

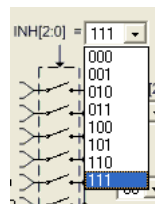


Figure 3-20



## INL Network

- (1) Click the network by mouse, INL can select the specified network.
- (2) Click the network switch by mouse, INL can select the specified network.
- (3) Click the mouse, a menu as Figure 3-21 will appear and users can select the switch network.

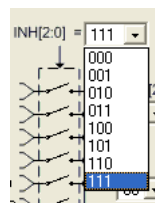


Figure 3-21

- INIS Switch
  - (1) Click the specified network by mouse, INIS switch will turn ON/OFF.
  - (2) Click the specified network switch by mouse, INIS switch will turn ON/OFF.
- INX Network Switch
  - (1) Click the specified network by mouse, 4 switches are shown as Figure 3-22.

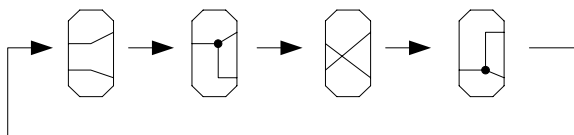


Figure 3-22

- (2) Click the mouse, a menu as Figure 3-23 will appear and users can select the switch network.

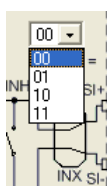


Figure 3-23

- INBUFF Switch
- Click the specified network by mouse, INBUF switch will turn ON/OFF.
- Click the specified network switch by mouse, INBUF switch will turn ON/OFF. INL network.
  - (1) Click the network by mouse, INL can select the specified network.
  - (2) Click the network switch by mouse, INL can select the specified network.
  - (3) Click the mouse, a menu as Figure 3-24 will appear, and users can select the specified switch network.

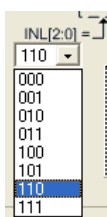


Figure 3-24

- ADC Sample Clock
 

Click the mouse, a menu as Figure 3-25 will show up, users can select the specified switch network.

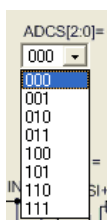


Figure 3-25

- ADCK Selection
  - (1) Click the specified network by mouse, ADCK will switch selection.

(2) Click the specified switch by mouse, ADCCK switch will switch selection.

➤ ENADC

Click the specified network by mouse, ENADC will turn ON/OFF. When ENADC = ON, display ADC zone will output value.

➤ VDDA Net work

ENVDDA enable control

(1) Select ENVDDA switch ON/OFF.

(2) Select VDDA voltage.

Click the mouse, a menu as Figure 3-26 will show up. Users can select to specified mode.

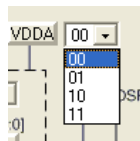


Figure 3-26

(3) Display VDDA Voltage

When ENVDDA = 0, the zone will show VDDA = External

When ENVDDA = 1, the zone will show VDDX selected voltage.

➤ PGA Network

Click the mouse, a menu as Figure 3-27 will appear. Users can choose the specified network.

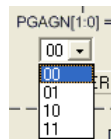


Figure 3-27

➤ VRH Network

(1) Click the network by mouse, VRH can select the specified network.

(2) Click the network switch by mouse, VRH can select the specified network switch. VRH can choose the specified network.

(3) Click the network by mouse, a menu as Figure 3-28 will appear. Users can select the specified switch network.

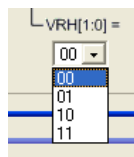


Figure 3-28

➤ VRL Network

(1) Click the network by mouse, VRL can select the specified network.

(2) Click the network switch by mouse, VRL can select the specified network.

(3) Click the mouse, a menu as Figure 3-29 will appear. Users can select the specified switch network.

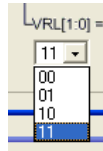


Figure 3-29

- VERBUFF
  - (1) Click the network by mouse, VERBUFF switch will turn ON/OFF.
  - (2) Click the network switch by mouse, VERBUFF switch will turn ON/OFF.
- DC Offset Network
 

Click the mouse, a menu as Figure 3-30 will show up. Users can select the specified network.

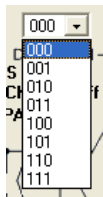


Figure 3-30

- ADGN Network
 

Click the mouse, a menu as Figure 3-31 will appear. Users can select the specified network.

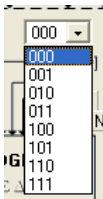


Figure 3-31

- VRGN
 

Click the network by mouse, VRGN can select the specified network.
- ENCHR
 

Click the network by mouse, ENCHR can select the specified network.

**Notice: ENCHR lightening up means ADC Chopper is closed.**
- ENHIGN
 

Click the network by mouse, ENHIGN can select the specified network.
- OSR Network
 

Click the mouse, a menu as Figure 3-32 will show up. Users can select the specified network.

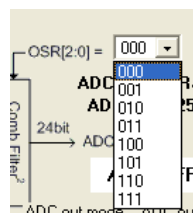


Figure 3-32

- ADC Display Zone
  - (1) Select ADC value output type → Hex or Dec output is selectable.
  - (2) Select ADC value output Bit → selectable 8 ~ 24 Bit output.
  - (3) Display output button → Click this button can immediately display ADC value.

## 3.7 OP Window

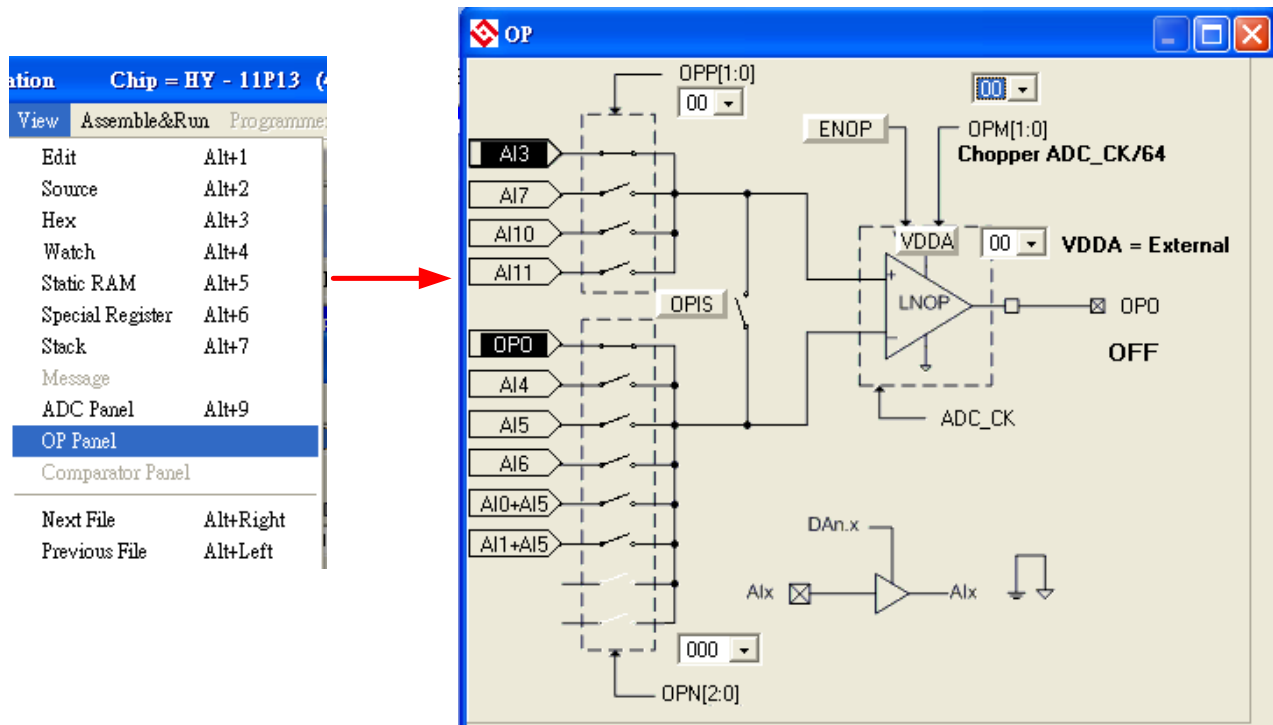


Figure 3-33

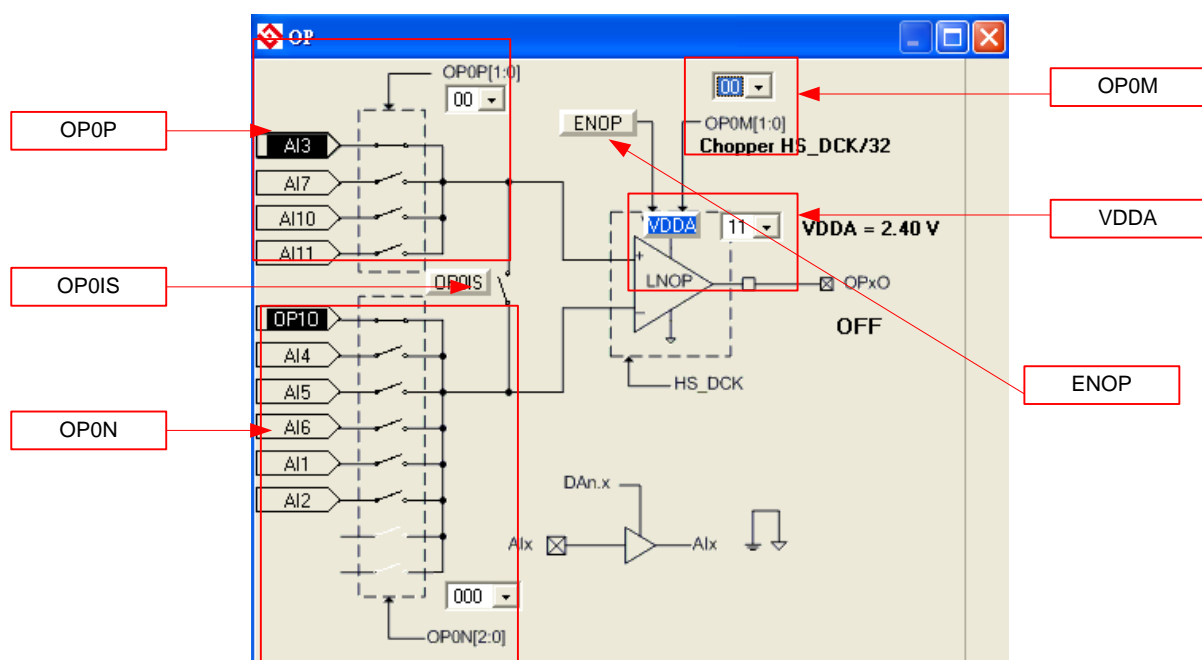


Figure 3-34



- OP0P Network
  - (1) Click the network by mouse, OP0P can select the specified network.
  - (2) Click the network switch by mouse, OP0P can select the specified network.
  - (3) Click the mouse, a menu as Figure 3-35 will appear. Users can select the specified switch network.

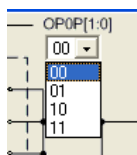


Figure 3-35

- OP0N Network
  - (1) Click the network by mouse, OP0N can select the specified network.
  - (2) Click the network switch by mouse, OP0N can select the specified network.
  - (3) Click the mouse, a menu as Figure 3-36 will appear. Users can select the specified switch network.

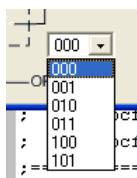


Figure 3-36

- ENOP

Click the network by mouse, ENOP will turn ON/OFF.

ENOP status display

When ENOP = 1, it will display ON

When ENOP = 0, it will display OFF
- VDDA Network (Please refer to VDDA network of ADC window)
- OP0M Network

Click the mouse, a menu as Figure 3-37 will show up. Users can select the specified switch network.

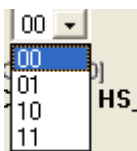


Figure 3-37

## 3.8 Comparator Window

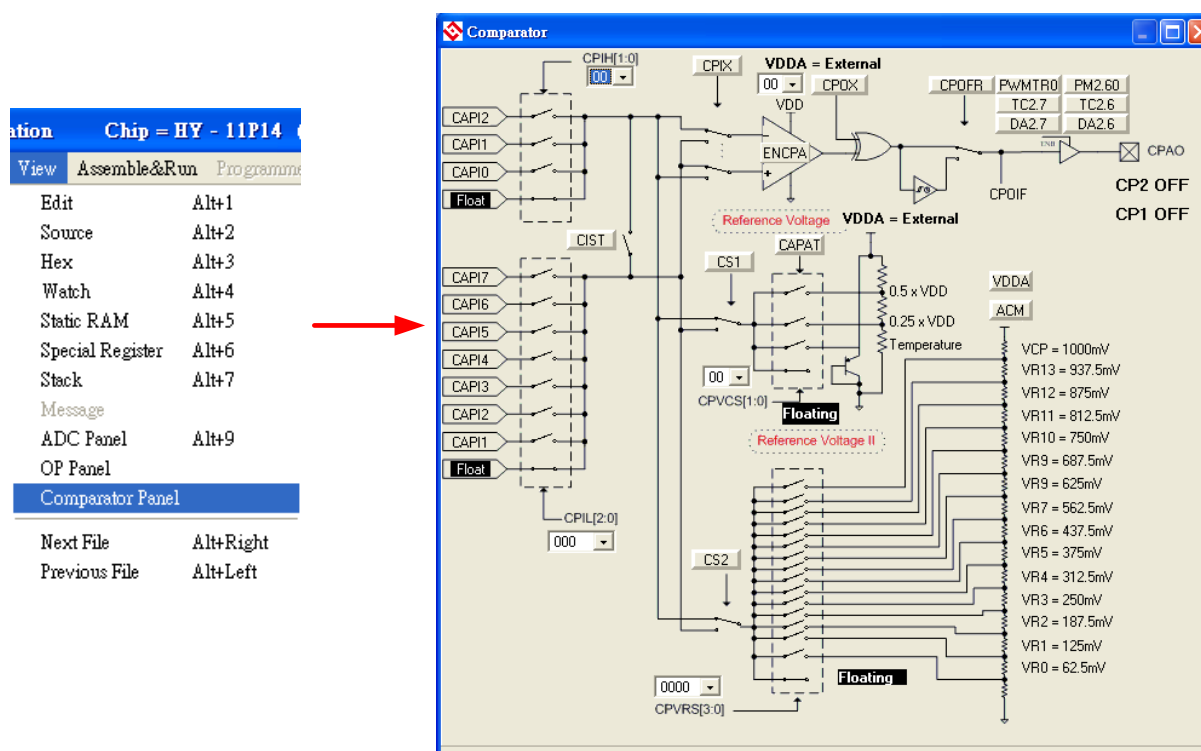


Figure 3-38

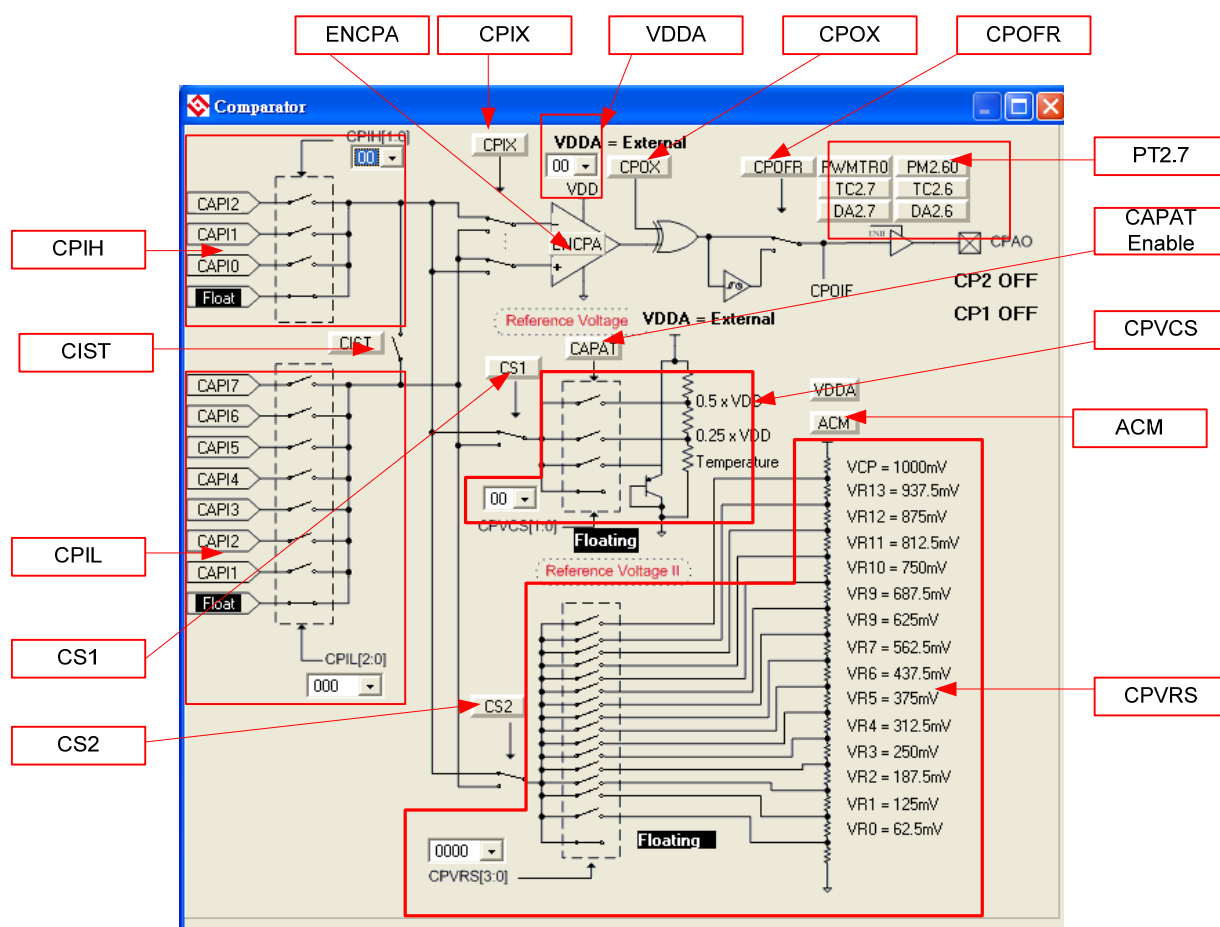


Figure 3-39

➤ CPIH Network

- (1) Click the network by mouse, CPIH can select to the specified network.
- (2) Click the network switch by mouse, CPIH can select to the specified network.
- (3) Click the mouse, a menu as Figure 3-40 will show up. Users can select the specified switch network.

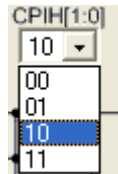


Figure 3-40

➤ CPIL Network

- (1) Click the network by mouse, CPIL is select to the specified network.
- (2) Click the network switch by mouse, CPIL is select to the specified network.
- (3) Click the mouse, a menu as Figure 3-41 will appear. Users can select to the specified switch network.

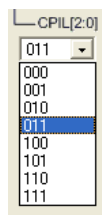


Figure 3-41

➤ CPVCS Network

- (1) Click the network by mouse, CPVCS can select to the specified network.
- (2) Click the network switch by mouse, CPVCS can select the specified network.
- (3) Click the mouse, a menu as Figure 3-42 will show up. Users can select the specified switch network.

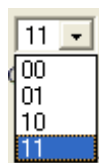


Figure 3-42

➤ CPVRS Network

- (1) Click the network by mouse, CPVRS can select to the specified network.
- (2) Click the network switch by mouse, CPVRS can select the specified network.
- (3) Click the mouse, a menu as Figure 3-43 will show up. Users can select the switch network.

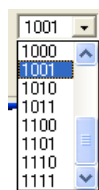


Figure 3-43

- VDDA Network (Please refer to the VDDA network of ADC window)
- CPIX Switch  
Click the network by mouse, CPIX will switch.
- CIST  
Click the network by mouse, CIST will turn ON/OFF.  
CIST status display:  
When CIST = 1, display ON  
When CIST = 0, display OFF
- CS1 Switch  
Click the network by mouse, CS1 will switch up and down.
- CS2 Switch  
Click the network by mouse, CS2 will switch up and down.
- ENCPA  
Click the network by mouse, ECPA will be activated.  
ENCPA status display:  
When ENCPA = 1, comparator is enabled.  
When ENCPA = 0, comparator is disabled.
- CPOX  
Click the network by mouse, CPOX will turn ON/OFF.
- CPOFR  
Click the network by mouse, CPOFR will switch up and down.
- CAPAT enable  
Click the network by mouse, CAPAT module will be activated or inactivated.  
CAPAT status display:  
When CAPAT = 1, CAPAT module is activated.  
When CAPAT= 0, CAPAT is inactivated.
- REFO Switch  
Click the network by mouse, REFO will turn ON/OFF.  
REFO status display:  
When REFO = 1, REFO is activated.  
When REFO = 0, REFO is inactivated.
- PT2.7  
CAPO will output only when PT2M.7 = 1, TC2.7 = 1 and DA2.7 = 1,.

## 3.9 Register & SRAM Revise Record

If the register or SRAM has been revised manually after access to emulation window (hardware emulation or software emulation), the data will be recorded (despite the RAM, Register, ADC, OP and CMP is revised through any kind of window). The data will be revealed after pressing the button "RAM revise record". At this time, windows will suspend until it is closed to execute other commands.

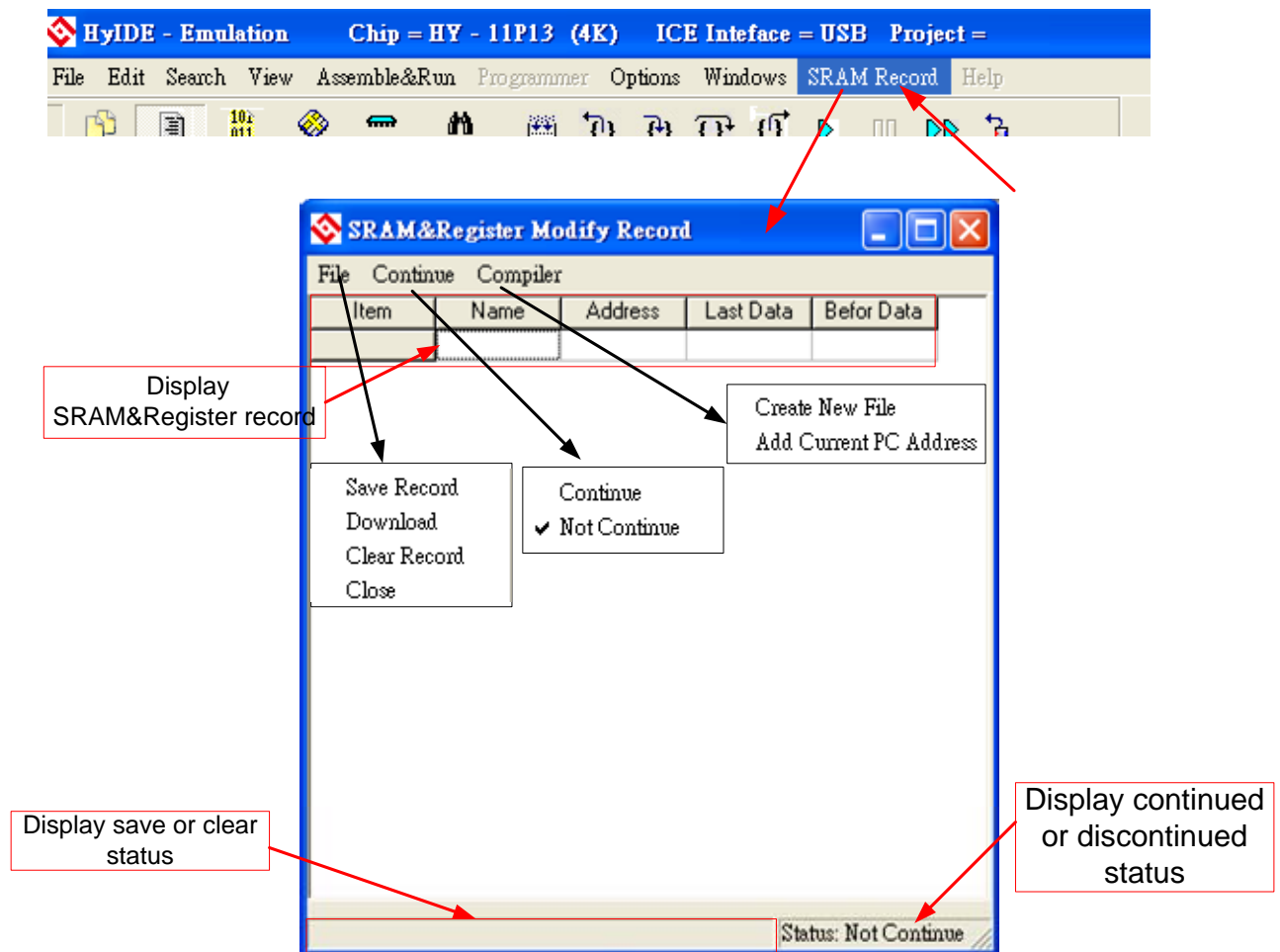


Figure 3-44

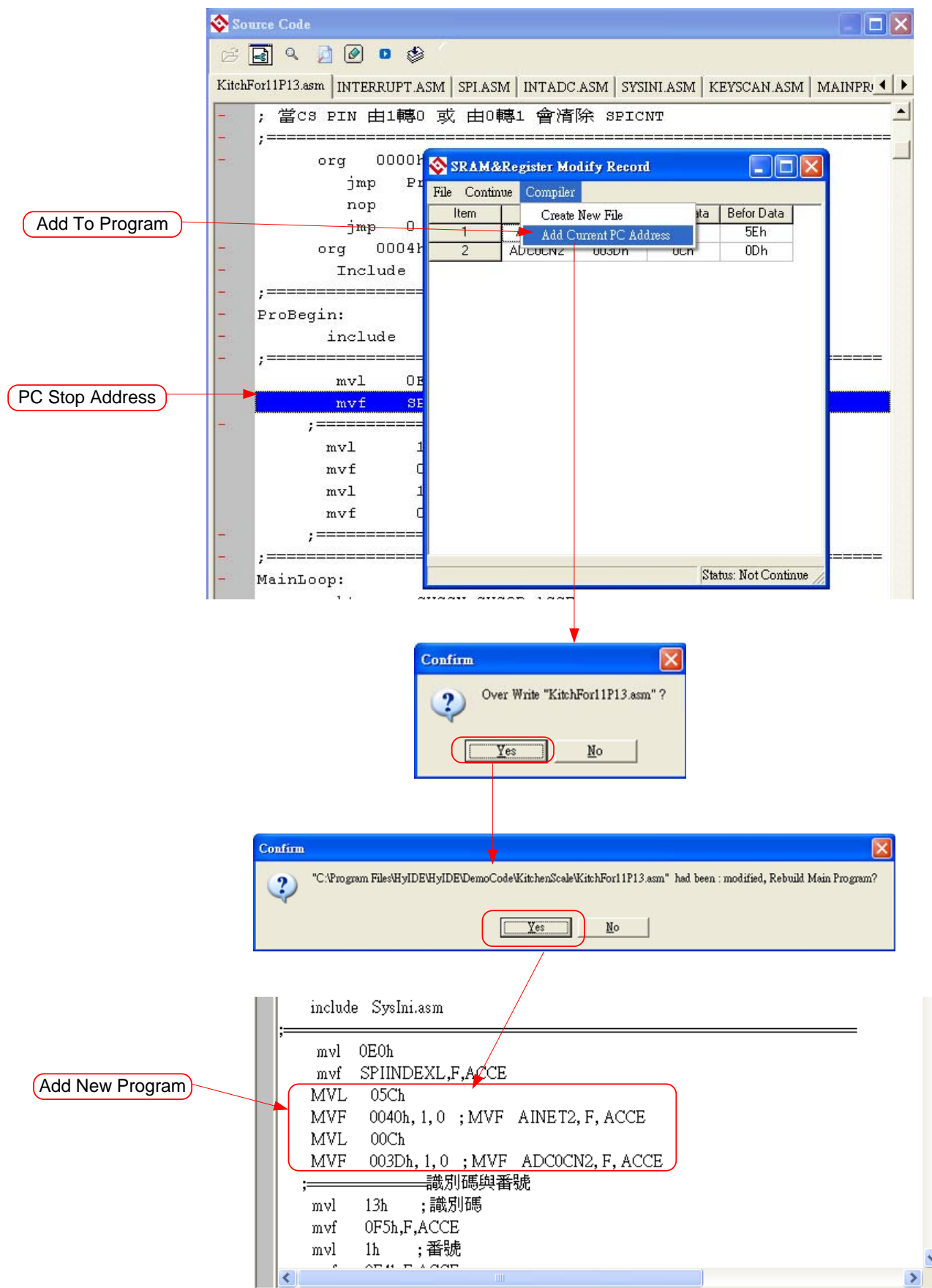


Figure 3-45

## 3.10 Hint Function of Source Code Window

If users intend to know Register or SRAM value and Address in source code window, point the cursor to register or SRAM, the name, address and data can be revealed.

This function is only applicable to the instructions below:

CLRF, ADDF, INF, INSZ, DCF, DCSZ, SUBF, COMF, ADDC, ANDF, IORF, XORF, SUBC, RRF, SETF, MULF, RLF, JZ, RRFC, RLFC, SWPF, DAW, INSUZ, DCSUZ, ARLC, ARRC, CPSG, CPSL, CPSE, TFSZ, BTFG BSF, BCF, BTSS, BTSZ, MVFF(not Macro).

- Only the first followed argument is effective as Figure 3-46 described.
- When command is BCF, BSF, BTSS, BTSZ and BTGF, Byte value will be revealed if the cursor points to the first argument. If the cursor points to the second argument, it will display the specified Bit value (1 or 0) as Figure 3-47 illustrated.
- When command is MVFF (not Macro), first argument value will appear if the cursor points to the first argument. If the cursor points to the second argument, argument value will show up as shown in Figure 3-48.
- If the argument is INDF0, POINC0, PODEC0, PRINC0, INDF1, POINC1, PODEC1 and PRINC1, the Data will be FSR0 or the address Data of FSR1 as
- Figure 3-49 described.
- If the argument is PLUSW0 or PLUSW1, the Data is FSR0+WREG or the address Data of FSR1+WREG as illustrated in Figure 3-50.

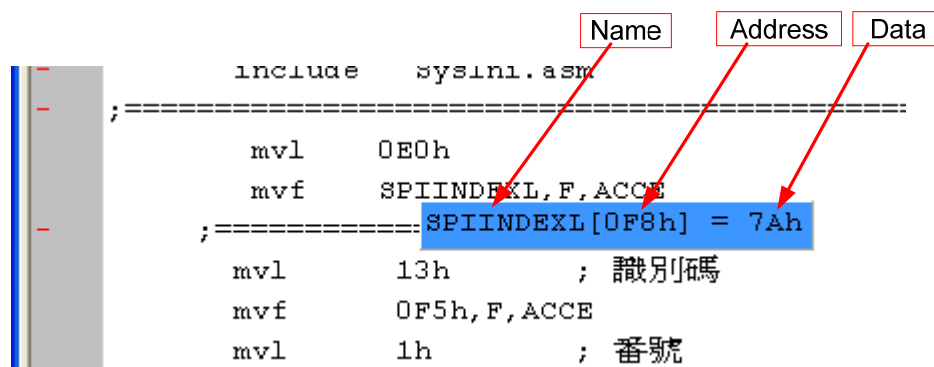


Figure 3-46

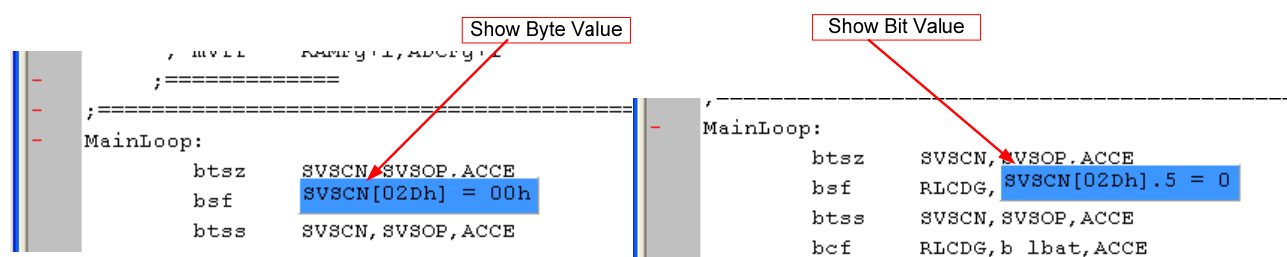


Figure 3-47

Figure 3-48 displays two assembly code snippets. The left snippet shows a `bsf` instruction with the first argument `INDF0,4,ACCE` highlighted in blue, with a red box labeled "First argument" pointing to it. The right snippet shows a `bsf` instruction with the second argument `ADCFG+1[0A9h] = 7Fh` highlighted in blue, with a red box labeled "Second argument" pointing to it.

```

-      bsf      INDF0,4,ACCE
-      ; AS 10,80h
-      mvff     RAMFG+1,ADCFG+1
-      ;===== RAMFG+1[0A5h] = FFh
-      ;=====
-      MainLoop:
-      btsz     SVSCN,SVSOP,ACCE

```

```

-      bsf      INDF0,4,ACCE
-      ; AS 10,80h
-      mvff     RAMFG+1,ADCFG+1
-      ;===== ADCFG+1[0A9h] = 7Fh
-      ;=====
-      MainLoop:
-      btsz     SVSCN,SVSOP,ACCE

```

Figure 3-48

Figure 3-49 displays an assembly code snippet. The `INDF0[120h] = FEh` line is highlighted in blue. Red boxes labeled "Name", "FSR0 address", and "Data" point to the `INDF0`, `120h`, and `FEh` respectively.

```

-      mvff     INDF1,PLUSW0
-      bsf      INDF0,4,ACCE
-      ; AS 10,80h INDF0[120h] = FEh
-      mvff     RAMFG+1,ADCFG+1
-      ;=====
-      ;=====
-      MainLoop:
-      btsz     SVSCN,SVSOP,ACCE
-      btsz     SVSCN,SVSOP,ACCE

```

Figure 3-49

Figure 3-50 displays an assembly code snippet. The `PLUSW0[145h] = A7h` line is highlighted in blue. Red boxes labeled "Name", "FSR0+WREG Address", and "Data" point to the `PLUSW0`, `145h`, and `A7h` respectively.

```

-      mvl      4
-      mvff     INDF1,PLUSW0
-      bsf      INDF0, PLUSW0[145h] = A7h
-      ; AS 10,80h
-      mvff     RAMFG+1,ADCFG+1
-      ;=====

```

Figure 3-50



## 4. Programming Window

### 4.1 Interface setting

Click "Option" button and select interface setting to get access to programming window as

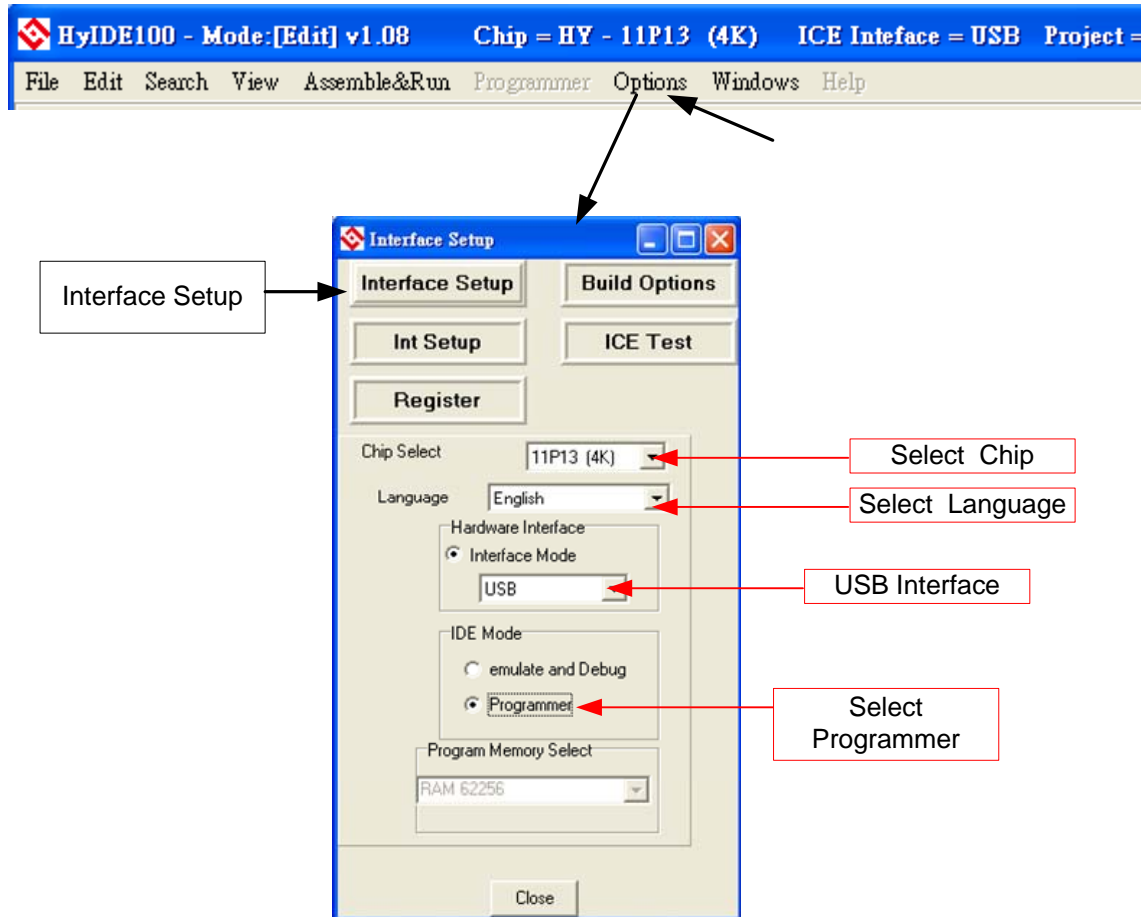


Figure 4-1 described.

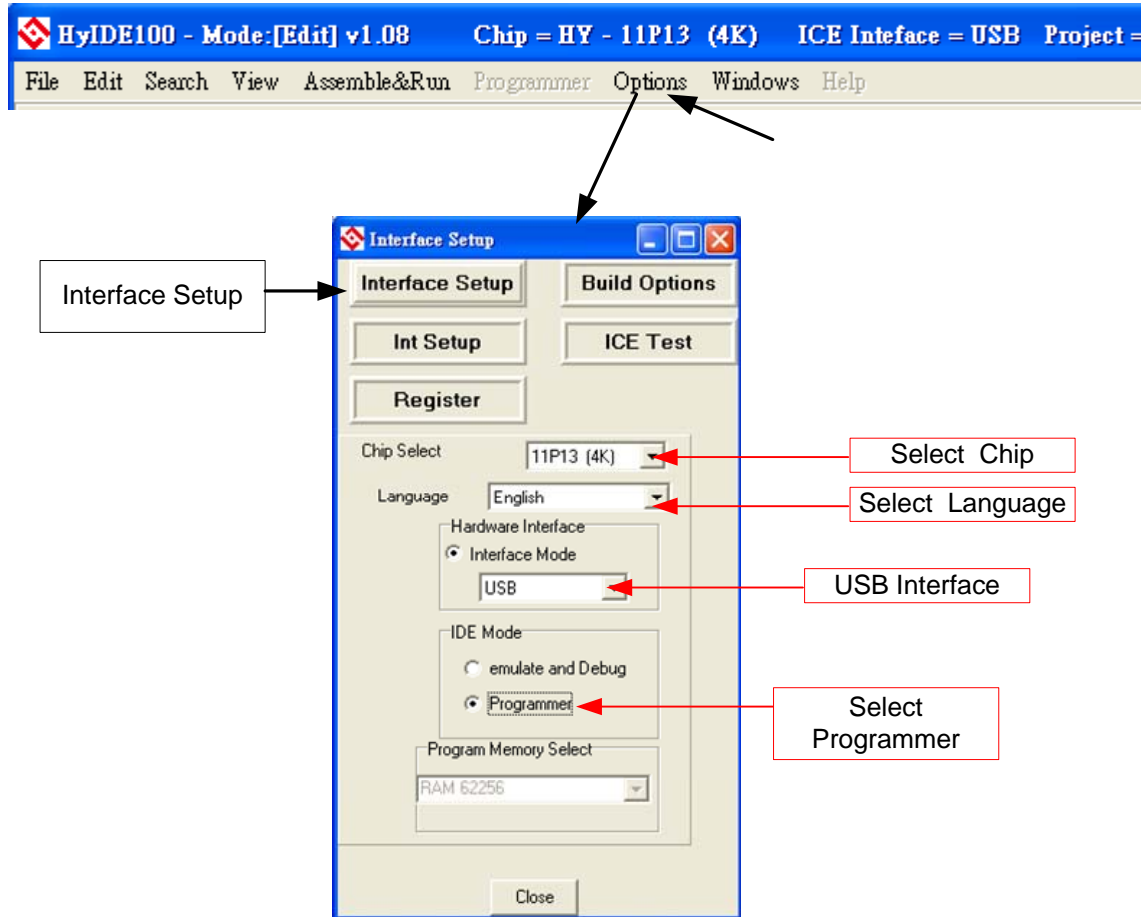


Figure 4-1

IC Selection → Select the IC part no. If the programming IC differs from the selected part no., Blank Check, Program and Verify will fail.

Language → Select the language of operating interface, either Chinese or English.

Hardware Setting → USB or Parallel Port interface is selectable.

IDE Mode → Select programming.

When the interface setting is accomplished, click "Assemble Option" to select programming setting as Figure 4-2 indicated.

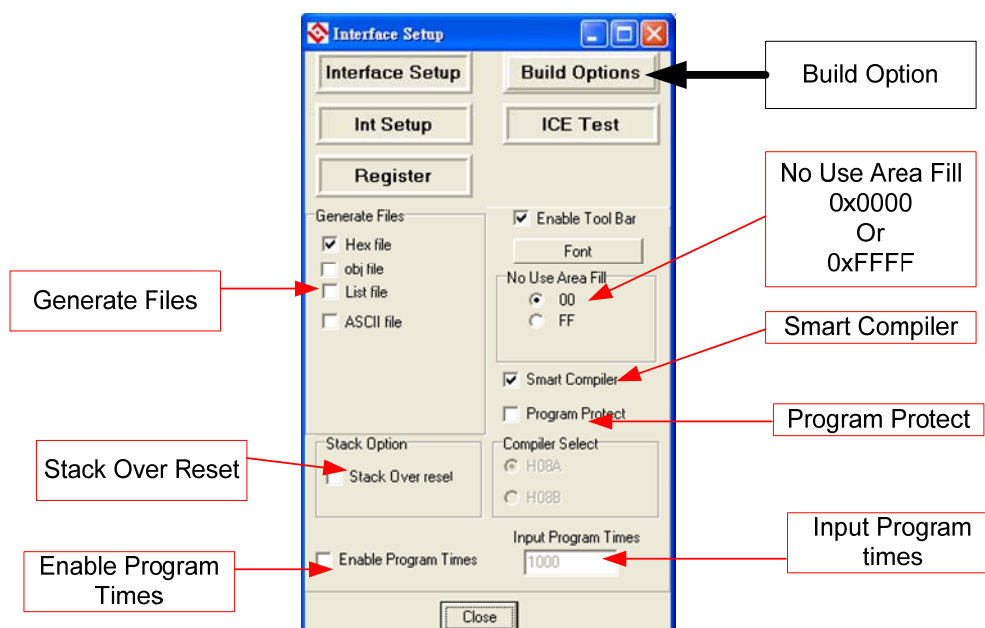


Figure 4-2

Assemble Generated Extension → The generated file after selecting the programming program.

Stack Operation → Select whether to replace when stack overflow/underflow occurred after OTP program operation.

Fill Unused Zone → Fill the unused zone with 00 or FF in the program after programmed.

Simplified Assemble → Select whether to simplify assemble.

Enable Programming Times → Select whether to enable Download program's programming times.

Input Programming Times → Fill in Download program's programming times. (Maximum is 2147483646, minimum is 1).

After assembling finished, click "ICE Test" to evaluate testing voltage as Figure 4-3 described (Connect IDE and insert 9V power before clicking"Option").

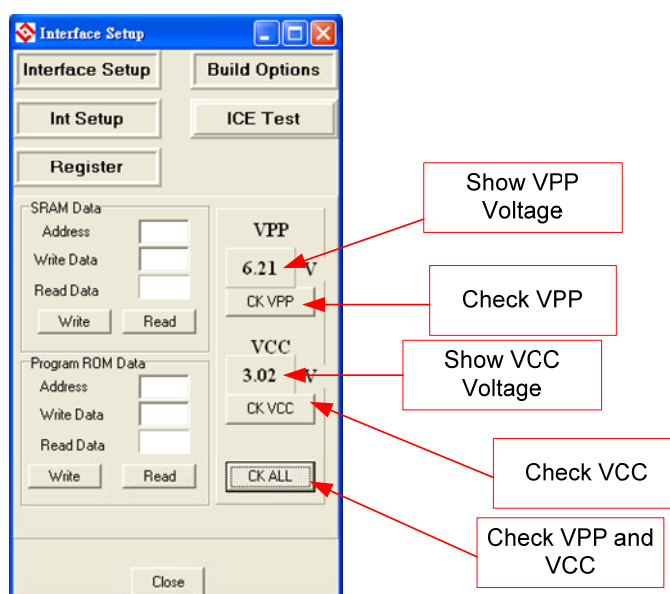


Figure 4-3

Click "Close" after the interface setting is done. All arguments will be recorded. When the setting is opened next time, default value will be written in automatically and the selected programming IC part no. will be shown in topic window as Figure 4-4 described.

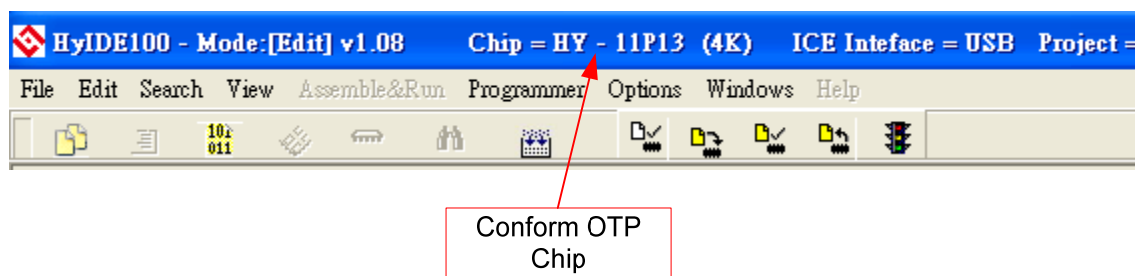


Figure 4-4

VPP voltage while programming:  $5.6 < VPP < 6.6$ .

VDD voltage while programming:  $2.7 < VPP < 3.6$ .

## 4.2 Operation Procedures

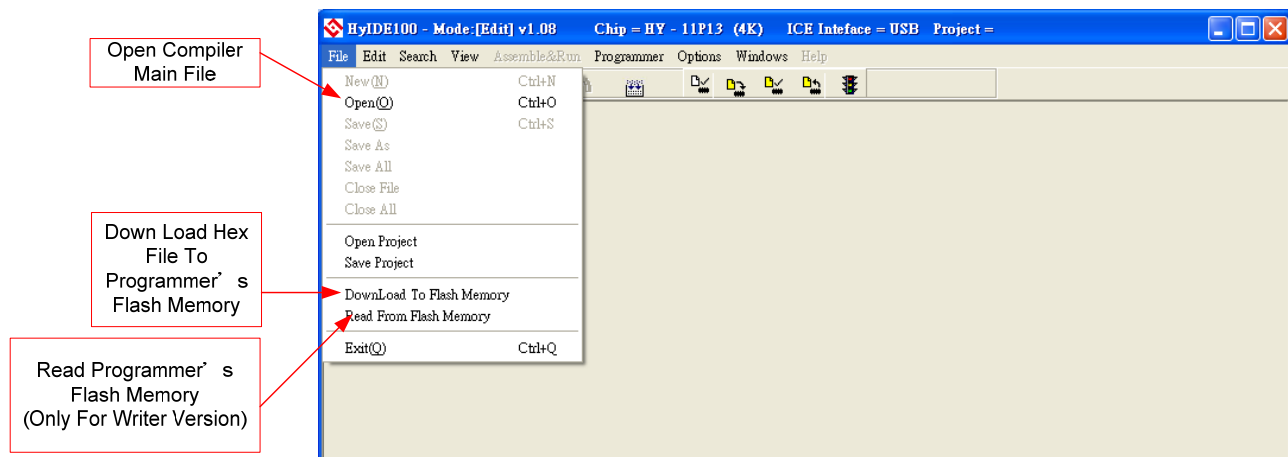


Figure 4-5

Open → Open the programmed source code main file.

Open Project → Open the saved project.

Save Project → Save the finished project.

Download file to Flash Memory → Download the finished Hex file after assembling to programmer or IDE Flash Memory.

## 4.2.1 Open File and Assemble

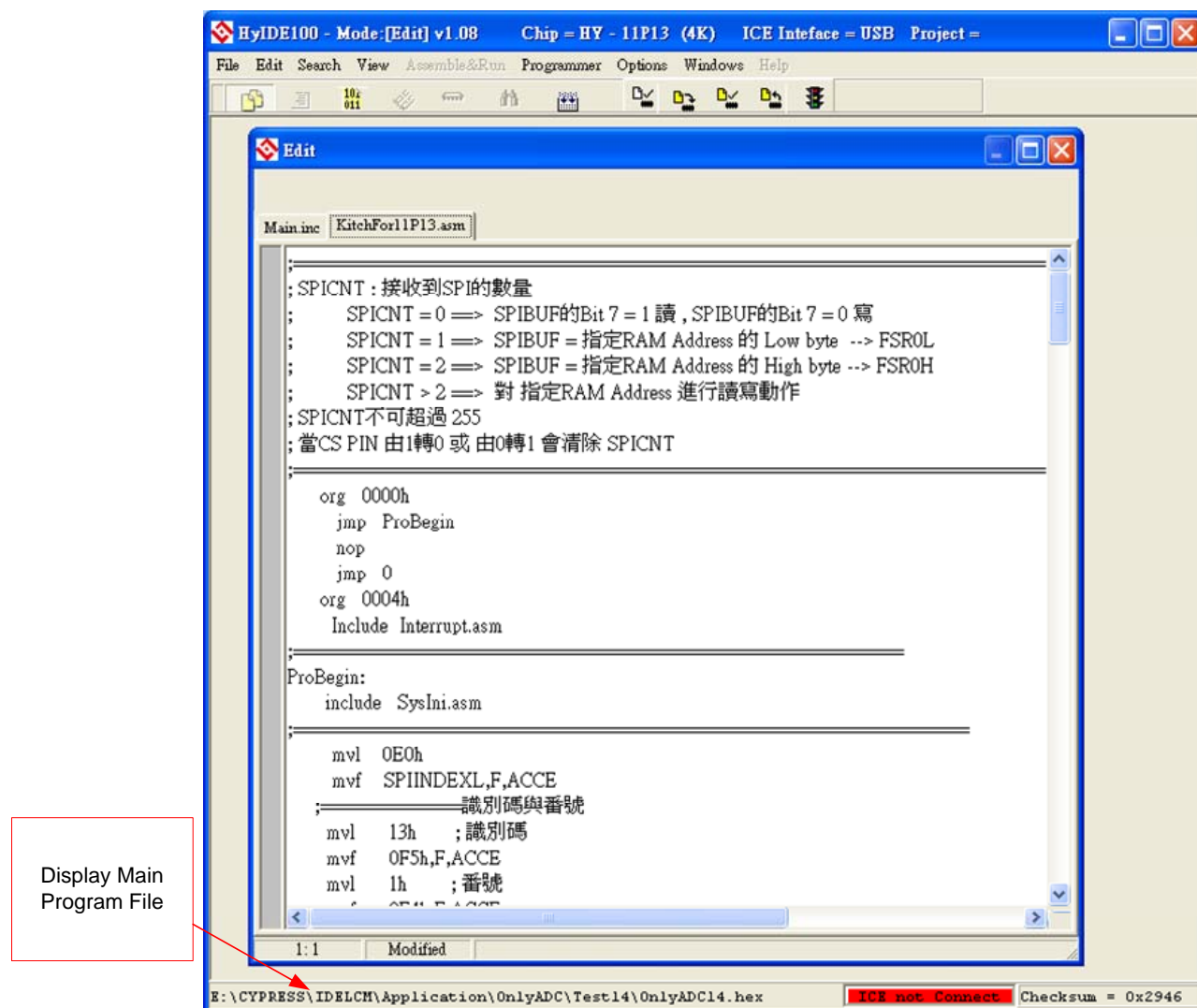


Figure 4-6

Open source code main file and display the assemble file. If the displayed name differs from main file, point the cursor to the file and press mouse right key. Set this file as the assembling main file as shown in Figure 4-7.

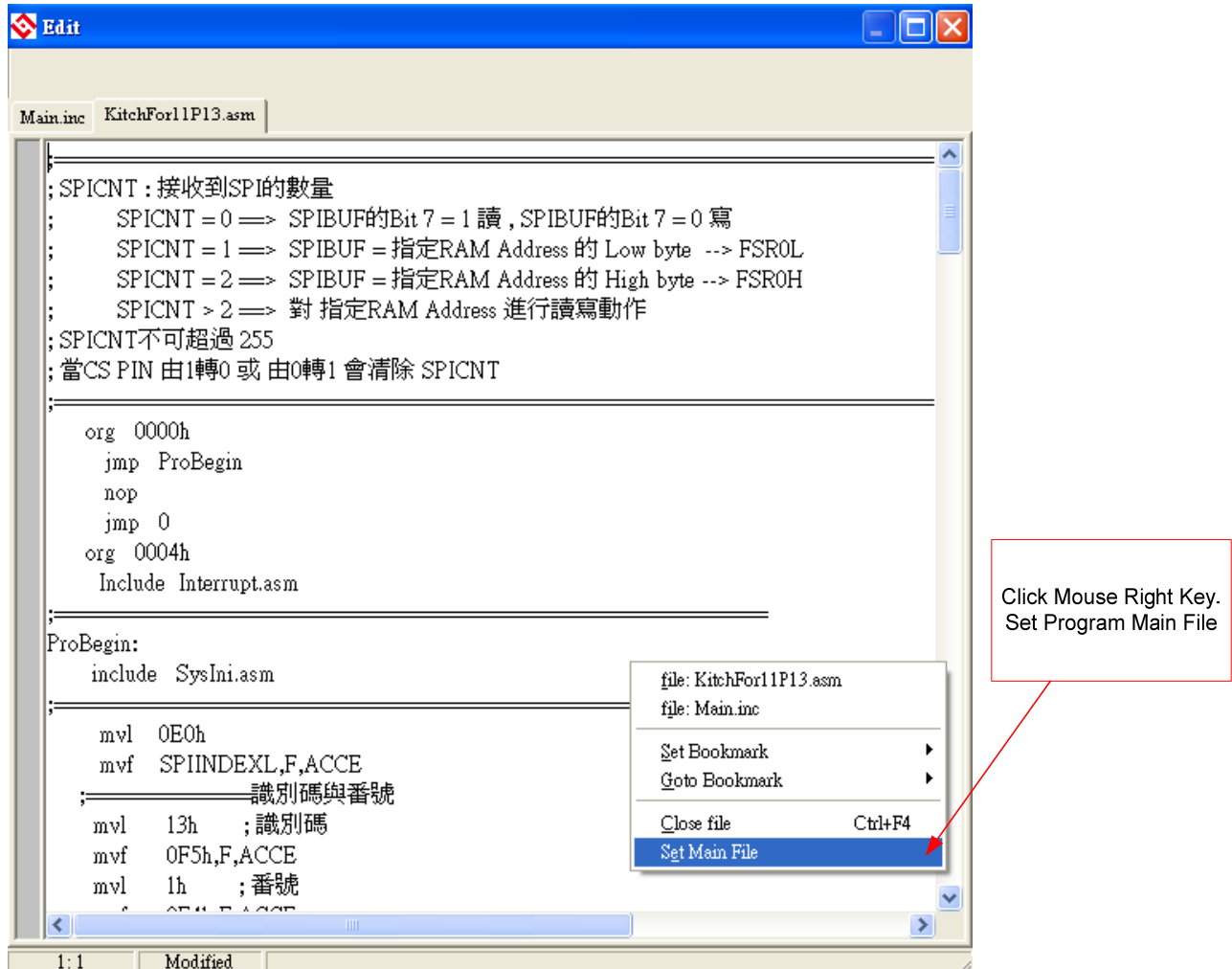


Figure 4-7

Assemble Source Code and Download it to programmer or IDE Flash Memory as Figure 4-8 described.

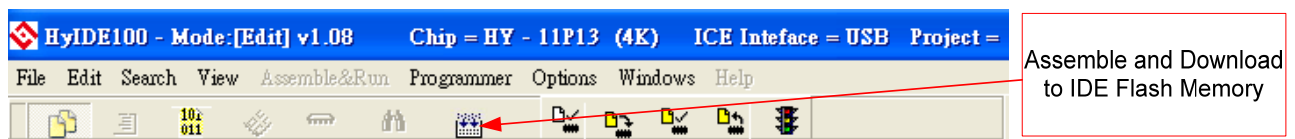


Figure 4-8

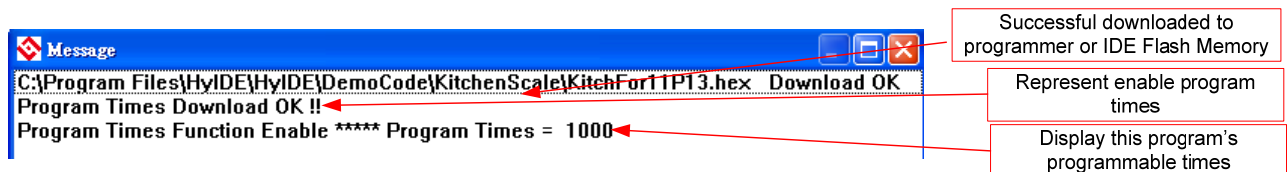


Figure 4-9

1. When using USB interface, the finished program code will be loaded into programmer or Flash Memory of IDE for mass production programming.
2. If there is programming times in the assemble option, information column will display the times of programming times as shown in Figure 4-9.

3. After assemble completed, Hex filename and Checksum will display in underneath part, as Figure 4-10 illustrated.

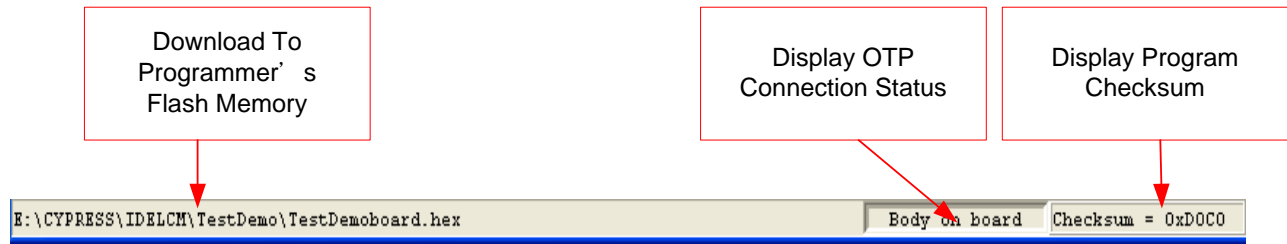


Figure 4-10

#### 4.2.2 Download Hex File to Programmer or IDE Flash Memory

To program an assembled Hex file, click “File”, select “Download to Flash Memory”. Choose the specific Hex file to conduct programming, as described in

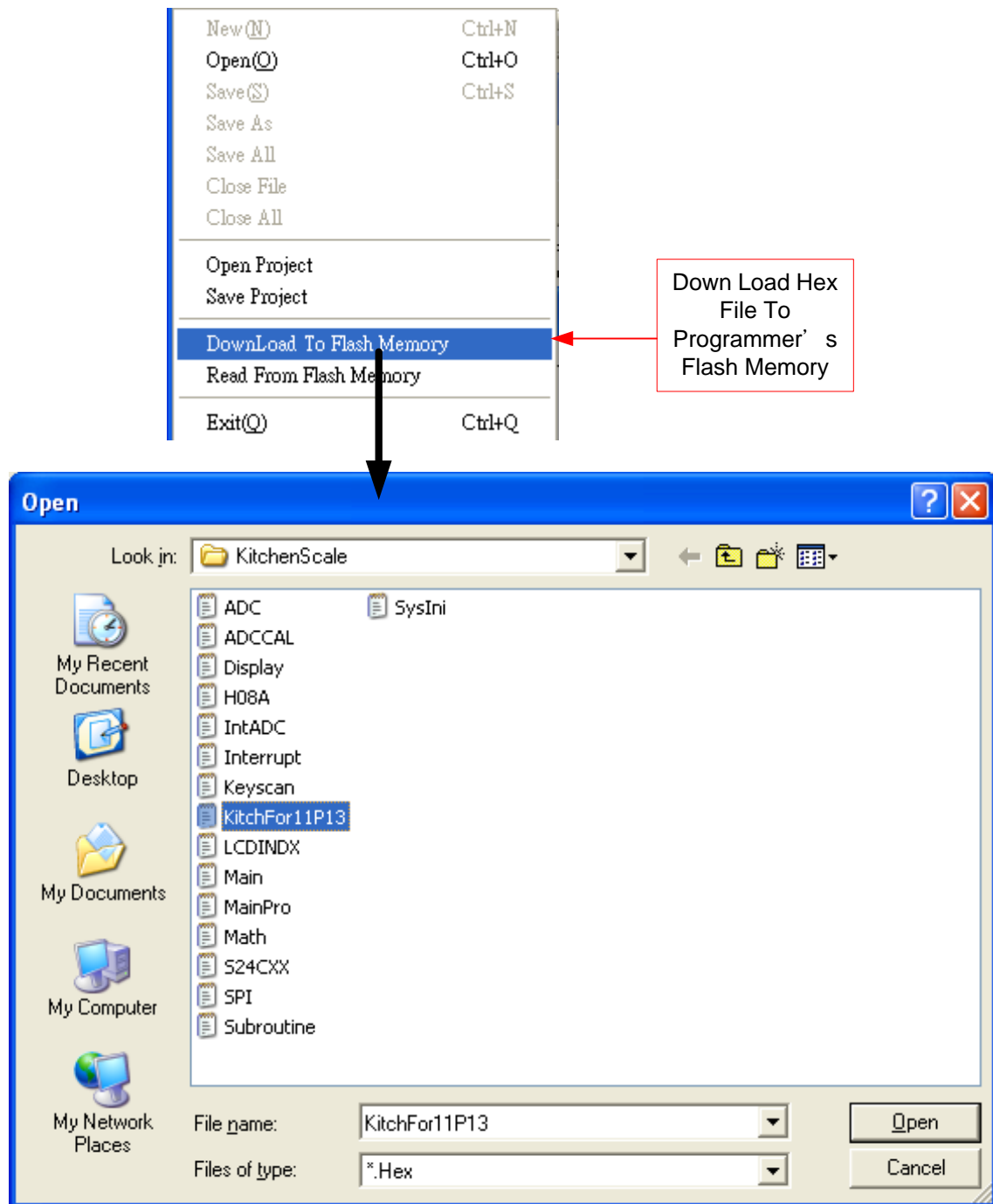


Figure 4-11.



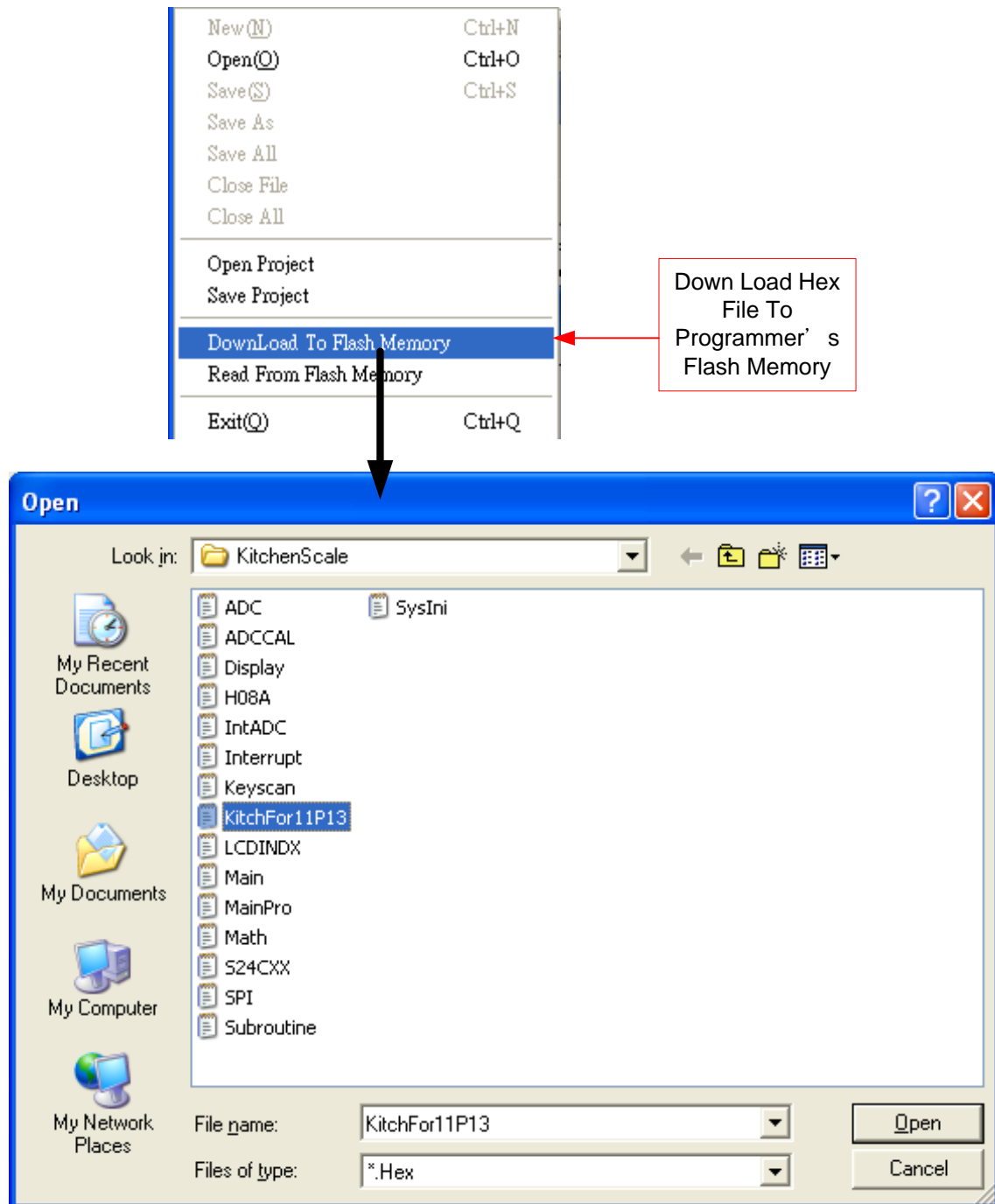


Figure 4-11

Information column will show success message after programming succeed, as shown inFigure 4-9. The Downloaded Hex file and Checksum will also be revealed in the indication column asFigure 4-10 illustrated.

### 4.3 PC Online OTP Programming

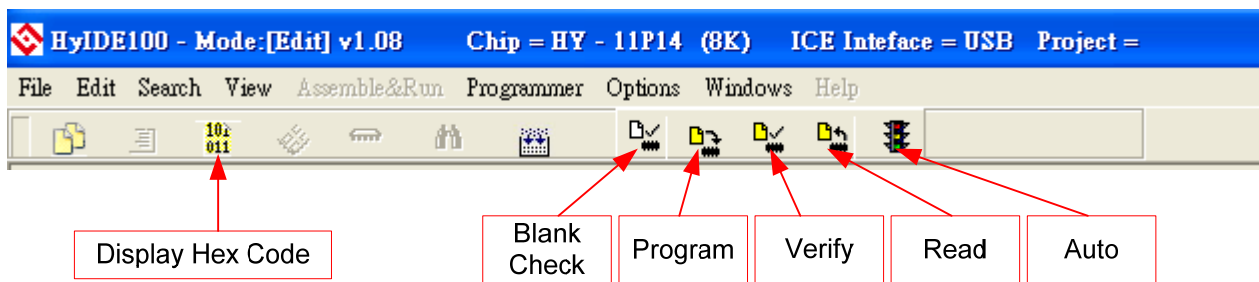


Figure 0-12

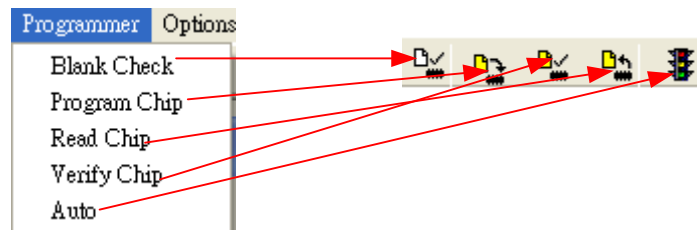


Figure 4-13

Blank Check, Programming, Verify and Read command can be implemented when the programmed file being successfully loaded into programmer or IDE Flash Memory. On the contrary those commands will not be activated if the download is malfunctioned.



Figure 4-14



Figure 4-15



Figure 4-16

Make sure the selected programming IC part no. is the same with the OTP part no. in topic window as Figure 4-4 described. When programmer executes Blank Check, Programming and Verify commands, Program will check whether the IC part no. and OTP programming part no. are identical. If the part no is different, the data will not be written into OTP and an error message will display in information column

as



Figure 4-14 described.

If users intend to find out whether the part no. is correct before programming, point the cursor to "IC Connection Status Display" and click the mouse left key. If the IC part no. is correct, a message will show up as Figure 4-15. If it is incorrect, the message will display as Figure 4-16. If "Enable Program Times" is ticked, the spare program times will display in the message column as illustrated Figure 4-17.



Figure 4-17

## 4.3.1 Blank Check

The internal code of Blank ICs that have not yet been programmed is 0xFFFF. The purpose of checking the IC is to assure the OTP address content is 0xFFFF.

If the IC selection is correct and the content is empty, a message will appear as Figure 4-18.

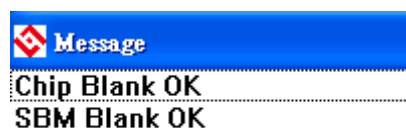


Figure 4-18

If the IC selection is incorrect or the content is not empty, a message will show up as Figure 4-19 described.

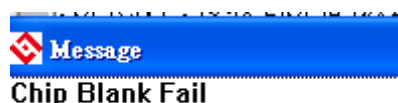


Figure 4-19

## 4.3.2 Program

The purpose of programming is to write Compiler accomplished program into IC OTP. When programming is completed and the IC is assembled as finished goods, it can operate the program as users commanded.

Program the downloaded or assembly finished Hex file (displayed at the bottom of the column) in the selected IC and verify the correctness of the programming content (please refer to Chapter 4.2.1 or 4.2.2 for programming procedures).

If the selected IC is correct and the programming succeeds, message will appear at the information column as Figure 4-20 illustrated. If "Enable Program Times" is ticked up, the enable program times will minus 1 and the remaining program times will be revealed in the message column.

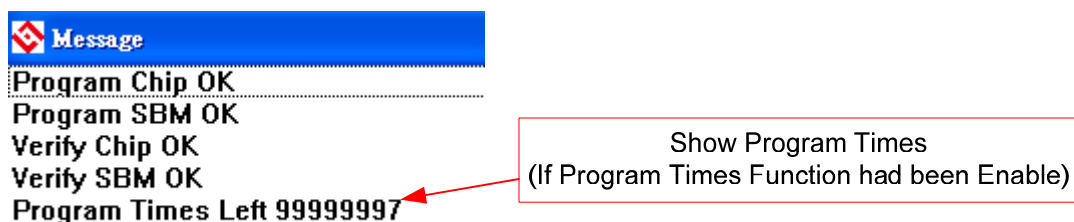


Figure 4-20

If the IC selection is incorrect or the programming fails, a message will show up as Figure 4-21).



Figure 4-21

### 4.3.3 Verify Program

The purpose to verify program IC is to compare if the program written into IC OTP is equal to the program downloaded to programmer or IDE Flash Memory.

Verify program IC content consistency with the downloaded or assembled Hex file (displayed at the bottom of the column). If the IC is protected by program, this verification is ineffective or the comparison fails.

If IC selection and program verification is success, a message will appear as Figure 4-22.



Figure 4-22

If IC selection is incorrect or the program verification miscarries, a message will pop up as Figure 4-23.



Figure 4-23

### 4.3.4 Read

The purpose to read the IC is to verify the consistency of OTP Checksum and programmed Hex

file. To read IC content, the procedures are illustrated as Figure 4-24. Its content will reveal at "Display Code" window.

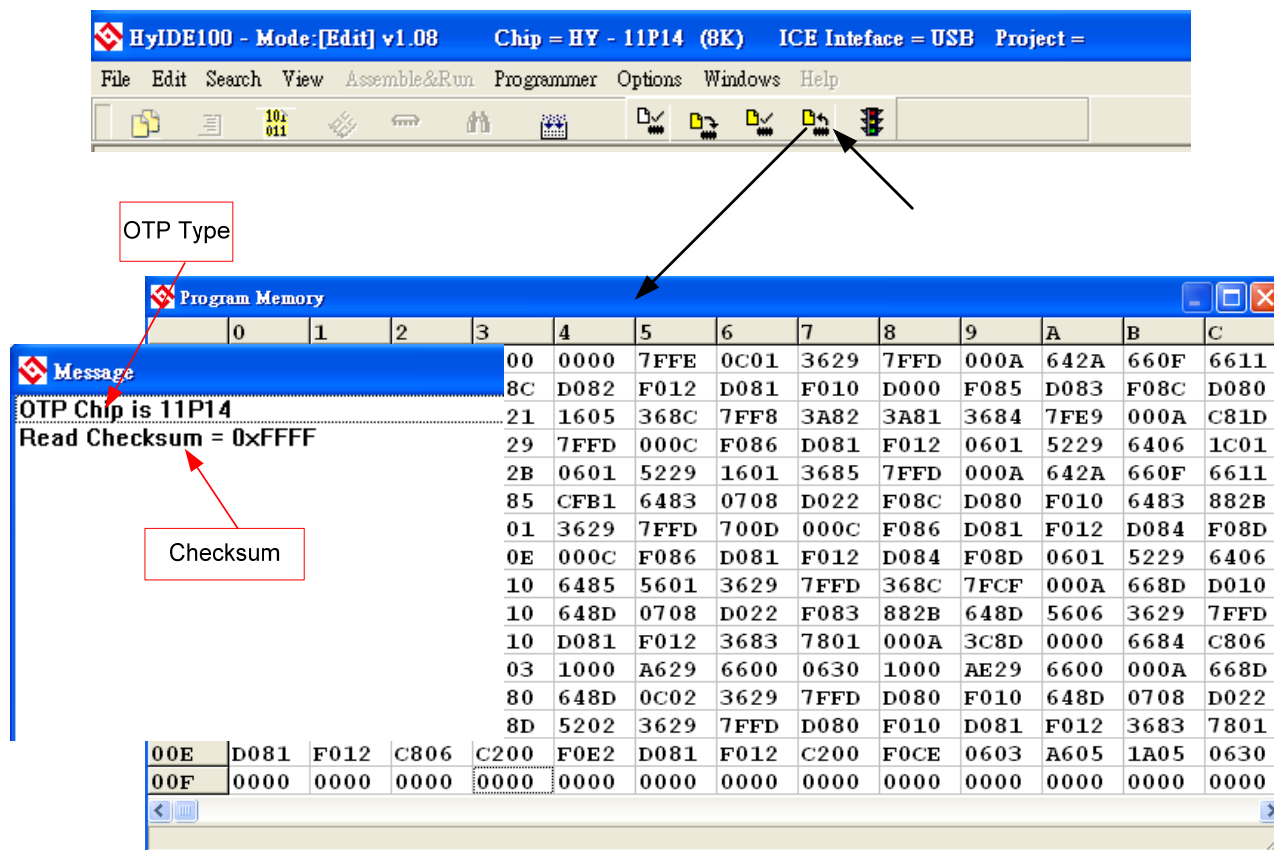


Figure 4-24

## 4.3.5 AUTO

Auto integrates the functions of Blank Check, Program and Verify. If user selects Auto, it will first check whether the IC is blank, then to program and verify.

After the execution succeeded, a message will be displayed as Figure 4-25 displayed. If the option, "Enable Program Times" is ticked up, the program permitted times will reduce 1 and the remaining program times will be shown in the message column.

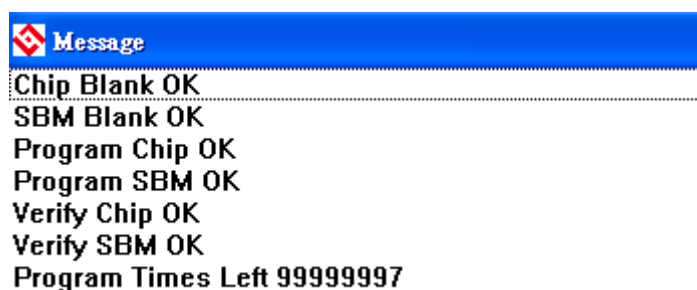


Figure 4-25

If any function fails, the whole process will stop and display an error message in the message column.

## 4.4 Offline Programming

### 4.4.1 Program Description

As the development process evolves to mass-production, the programmer can be used alone when programming on the production line. There is not necessary to connect the programmer to the PC.

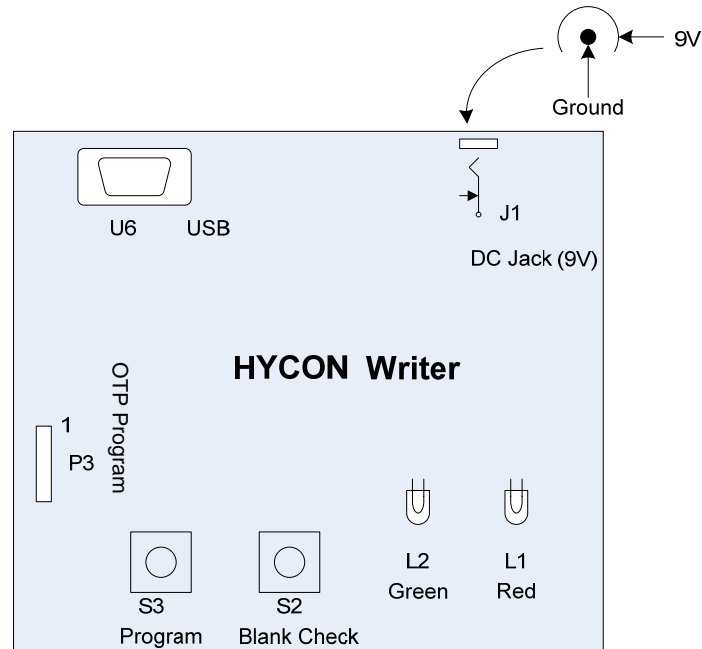


Figure 4-26

J1 → DC Jack, 9V input.

Notice: the case is 9V; the core is Ground (VSS).

U6 → USB port. Users use this port to connect to PC.

P3 → Program PIN output. PIN and OTP is connected.

PIN1 = VPP  
PIN2 = SCK  
PIN3 = DI  
PIN4 = DO  
PIN5 = VSS  
PIN6 = VDD

L1 → Error message light display, red LED.

L2 → Success message light display, green LED.

S2 → Blank Check button, for offline operation.

S3 → Program button, for offline program operation.

To implement offline operation, Hex file must be firstly downloaded to programmer Flash Memory. Please refer to Chapter 4.2.1 or 4.2.2 for procedures.

S2 Button can check if the IC is blank.

S3 Button is programming button. Its procedures are: Blank Check → Program → Verify. If "Program Protection" of "Assemble Option" is ticked up before downloading data to Flash Memory, program protection will be executed after Verify completed. If "Program Protection" is

not ticked up, it will stop after Verify accomplished.

If any failure or error happened during execution procedures, L1 red LED will be lightened up. On the contrary, L2 green LED will be lighted up if success.

## 4.4.2 Program Times Restriction

The menu of "Assemble Option" in interface setting has an option of "Enable Program Times" as described Figure 4-2. This option restrict the times of the permitted program times of download program.

This is a safety mechanism that restrains permitted program times, preventing it from over-programming on the production line.

After ticking up "Enable Program Times", key in program times in the filed below "Input Program Times" (maximum is 99999999, minimum is 1). This argument will be written into EEPROM of the programmer after the compiler programmed file is downloaded to Flash Memory. Afterwards, the enabled program times will reduce 1 each time when programming completed. If the value reduced to 0, the programming action may not be executed. At this time, an error signal (red LED) will be lighted up and Blank Check still operates normally.